```
PAGE
                                                           118,121
                                               TITLE Victor V86P Hard Disk Controller BIOS (2793VG.10010688)
                                                           INT 13H Commands
       = 0000
                                               INT13H OP 00 RESET DISK SYSTEM
                                                                                                                      00h
      = 0000
= 0008
                                               INT13H_OP_01_GET_DISK_STATUS
INT13H_OP_08_GET_DRIVE_PARAMS
                                                                                                          EQU
EQU
                                                                                                                      01h
                                                                                                                      08h
                                               INT13H_OP_09_INITIALIZE_DISK_TABLE
11
       = 0009
                                                                                                          EQU
                                                                                                                      09h
       = 0000
                                               INT13H_OP_OD_ALTERNATE_DISK_RESET
INT13H_OP_10_TEST_DRIVE_READY
                                                                                                          EQU
14
15
16
17
       = 0011
                                               INT13H_OP_11_RECALIBRATE
                                                                                                          EQU
                                                                                                                      11h
                                               INT13H_OP_12_SRAM_DIAGS
INT13H_OP_14_CONTROLLER_DIAGS
       = 0012
                                                                                                                      12h
       = 0014
                                                                                                           EQU
                                               LAST_INT13H_OP EQU
                                                                                 INT13H_OP_14_CONTROLLER_DIAGS
18
19
20
21
                                                          INT 13H Errors (returned in AH)
22
23
      = 0000
= 0001
= 0002
24
                                               INT13H STATUS 00 NO ERROR
                                                                                                                      000h
                                               INT13H_STATUS_01_BAD_COMMAND
INT13H_STATUS_02_ADDR_MARK_NOT_FOUND
                                                                                                                      001h
002h
                                                                                                          EQU
27
       = 0004
                                               INT13H STATUS 04 SECTOR NOT FOUND
                                                                                                          EOU
                                                                                                                      004h
                                               INT13H_STATUS_07_BAD_DISK_PARAM_TABLE
INT13H_STATUS_09_DMA_ACROSS_64K
28
       = 0007
                                                                                                          EQU
                                                                                                                      007h
29
                                                                                                                      009h
       = 0009
                                                                                                          EQU
                                                                                                          EQU
EQU
30
       = 000A
                                               INT13H STATUS OA BAD SECTOR FLAG
                                                                                                                      00Ah
                                               INT13H_STATUS_0B_BAD_CYLINDER
INT13H_STATUS_10_ECC_ERROR
31
32
       = 000B
                                                                                                                      OORh
          0010
                                                                                                                      010h
                                                                                                          EQU
       = 0011
                                               INT13H_STATUS_11_ECC_FIXED
                                                                                                          EQU
                                                                                                                      011h
33
34
35
36
         0020
                                               INT13H_STATUS_20_CTRLR_ERROR
INT13H_STATUS_27_NEED_RECALIBRATE
                                                                                                          EQU
EQU
                                                                                                                      020h
                                               INT13H_STATUS_28_UNKNOWN_ERROR
       = 0028
                                                                                                          EQU
                                                                                                                      028h
                                                                                                                                  ; Unknown bad sector error.
                                               INT13H_STATUS_40_SEEK_FAILURE
INT13H_STATUS_80_TIMEOUT
INT13H_STATUS_BB_UNDEFINED_ERROR
37
38
39
                                                                                                          EQU
EQU
       = 0040
                                                                                                                      040h
       = 00BB
                                                                                                          EQU
40
       = 00FF
                                               INT13H STATUS FF SENSE OP FAILED
41
42
43
44
45
46
                                                           Interrupt vector table (at 0000:0000) and BIOS Data Area (from 0000:0400)
                                                                                              SEGMENT AT 0000H
49
                                                                                              ORG
                                                                                                          013H*4
                                                                                                                      ; The disk services. We hook this
52
       004E ????
                                               INT13H SEGMENT
                                                                                              DW
                                                                                                                          and move the original handler over
53
54
55
                                                                                                                          to INT40H.
       0064
                                                                                              ORG
                                                                                                          019H*4
56
57
58
       0064 ????
0066 ????
                                                                                                                      ; Boot service. We hook this in order
; to try reading and running the
; hard disk boot sector first, then
                                               INT19H OFFSET
59
                                                                                                                          falling back to the floppy.
60
61
       0100
                                                                                              ORG
                                                                                                          040H*4
       0100 ????
                                                                                                                     ; The original (floppy) disk service ; is relocated here.
62
                                               INT40H OFFSET
                                                                                              DW
63
64
       0102 ????
65
       0104
                                                                                              ORG
                                                                                                          041H*4
66
67
       0104 ????
0106 ????
                                               INT41H_OFFSET
INT41H_SEGMENT
                                                                                              DW
DW
                                                                                                                      ; Data pointer, not code. Points to ; disk geometry and parameters.
68
       0442
0442 ??
0443 ??
69
70
71
72
73
74
75
76
77
                                                                                                          0442h
                                               BDA_CONTROLLER_DATA_BUFFER_00
                                                                                              DB
                                                                                                                      ; This is the buffer used for
                                                                                                                        constructing commands for submission into the disk controller, but also for reading the sense data on error. When a INT13H function returns, the first byte here allways holds the success/error status.
                                               BDA CONTROLLER DATA BUFFER 01
                                                                                              DB
                                               BDA_CONTROLLER_DATA_BUFFER_02
BDA_CONTROLLER_DATA_BUFFER_03
BDA_CONTROLLER_DATA_BUFFER_04
       0444 ??
0445 ??
                                                                                              DB
       0446 ??
                                                                                              DB
       0447 ??
                                               BDA CONTROLLER DATA BUFFER 05
                                                                                              DB
                                               BDA_CONTROLLER_DATA_BUFFER_06
78
79
80
       046C
                                                                                              ORG
                                                                                                          046Ch
       046C ????
046E ????
                                               BDA_TIMER_COUNTER_LO
BDA_TIMER_COUNTER_HI
                                                                                              DW
81
82
83
84
       0472
0472 ????
                                                                                              ORG
                                                                                                          0472h
                                               BDA_SOFT_RESET_FLAG
                                                                                              DW
       0474 ??
                                               BDA_LAST_OP_STATUS
BDA_NUMBER_OF_HARD_DISKS
                                                                                              DB
85
86
87
       0475 ??
                                                                                              DB
       7000
                                                                                              ORG
                                                                                                          7C00H
88
       7C00
                                               BOOT_SEC
                                                                                               LARET.
89
                                                                                                           7DFEh
       7DFE
                                                                                               ORG
                                               BOOT_SIG
                                                                                              LABEL
90
       7DFE
                                                                                                          FAR
91
92
       7DFE
93
94
       0000
                                               BIOSSEG
                                                                                              SEGMENT AT DECOUR
95
96
97
                                                                                               LABEL
       FFF0
                                               RESET_VEC
                                                                                                          FAR
       FFFO
                                               BIOSSEG
                                                                                              ENDS
```

100			PAGE								
101			;						_		
102			; Variables and buffers u				r		:		
103			; utility. The segment is						:		
104 105			;						-		
	0000		FORMATTER_HEAP	SEGMENT	AT 1000F	ī					
107											
	00B0			ORG	00B0h						
	00B0 00B1		SAVED_COLUMN SAVED_ROW	DB DB	?	;	DC	OKK	eepın	g routi	nes
	00B1		READ_NUMBER_SAVED_AL	DB	?						
	00B3		READ_NUMBER_SAVED_AH	DB	?						
			PARSE_NUMBER_BUFFER	DD	?						
	00B8		READ_NUMBER_RESULT	DW	?						
	OOBA	??	READ_NUMBER_RESULT_LEN	DB	?						
116 117	01B9			ORG	01B9h	,	For	rmat	ter v	ariable	s
	01B9	??	HARD_DISK_NUMBER	DB	?	′	- 0-			<u> </u>	•
	01BA		READ_NUMBER_MIN_DIGITS	DB	?						
	01BB		READ_NUMBER_MAX_DIGITS	DB	?						
	01BC			DB	?						
	01BD		DAD DOTMERD	DB	?						
	01BE 01BF		BAD_POINTER	DB DB	?						
	01C0		CURSOR_LOCATION_OR_SOMETHING	DB	?						
	01C1			DB	?						
127	01C2	??		DB	?						
	01C3			DB	?						
	01C4			DB	?						
	01C5 01C6			DB DB	?						
	01C7			DB	?						
	01C8		LAST_CYLINDER_NUMBER	DW	?						
	01CA			DB	?						
	01CB			DB	?						
	01CC		LAST_HEAD_NUMBER	DB	?						
	01CD		VERIFY_PROGRESS_BUFFER	DB DW	?						
	01CE 01D0		VERIFY_PROGRESS_BUFFER_0	DW	?						
	01D2		VERIFY_PROGRESS_BUFFER_1	DW	?						
	01D4		UNK_UNUSED_STH	DB	?						
	01D5		CONTROL_BYTE	DB	?						
	01D6		VERIFY_STH_2	DB	?						
	01D7		INTERLEAVE	DB	?						
	01D8 01D9			DB DB	?						
	01DA		CYLINDER_NUMBER	DW	?						
	01DC		HEAD_NUMBER	DB	?						
	01DD		_	DB	?						
	01DE			DB	?						
	01DF			DB	?						
	01E0 01E1			DB DB	?						
	01E1			DB	?						
	01E3		CURRENT_BAD_ENTRY	DW	?						
	01E5		CURRENT_BAD_ENTRY_0	DB	?						
	01E6		CYLINDER_TABLE	DB	?						
	01E7			DB	?						
	01E8			DB DB	?						
	01E9 01EA		BAD_BLOCKS_TABLE_USED_SPACE	DB	?						
	01EB		NUM_BAD_BLOCK_ENTRIES	DB	?						
	01EC		UNK_VERIFY_STH	DW	?						
	01EE		UNK_VERIFY_STH_0	DB	?						
	01EF		UNK_VERIFY_UNUSED_1	DW	?						
	01F1			DB	?						
	01F2 01F3			DB DB	?						
	01F4			DB	?						
	01F5			DB	?						
171	01F6	??		DB	?						
	01F7			DB	?						
	01F8		UNK_VERIFY_UNUSED_0	DW	?						
	01FA	7777	UNK_VERIFY_UNUSED	DW	?						
175 176											
	0400			ORG	0400h						
	0400	??	BAD_BLOCKS_TABLE	DB	?						
179											
	0700	22	GROWN DWEETER	ORG	0700h						
	0700	77	SECTOR_BUFFER	DB	?						
182 183	0701		FORMATTER_HEAP	ENDS							
184											
185											

186		PAGE				
187		;				
188 189		; Contro	ller opcodes			:
190		;				
	= 0000	OP_00_TEST_DRI	E READY		EQU	000h
	= 0001	OP_01_RECALIBR			EQU	001h
	= 0003	OP_03_READ_SEN			EQU	003h
194	= 0004	OP_04_FORMAT_D	RIVE		EQU	004h
195	= 0005	OP_05_VERIFY_S			EQU	005h
	= 0006	OP_06_FORMAT_T			EQU	006h
	= 0007	OP_07_FORMAT_B			EQU	007h
	= 0008	OP_08_READ_SEC			EQU	008h
	= 000A = 000B	OP_OA_WRITE_SEC	CTORS		EQU	00Ah 00Bh
	= 000B	OP_OC_INIT_DRV	DADM		EQU	00Ch
	= 000D		_BURST_ERROR_LE	NT	EQU	00Dh
	= 000E	OP_OE_READ_SEC		.•	EQU	00Eh
	= 000F	OP_OF_WRITE_SE			EQU	OOFh
	= 00E0	OP_EO_SECTOR_B			EQU	0E0h
	= 00E3	OP_E3_DRIVE_DI			EQU	0E3h
	= 00E4	OP_E4_CTRL_DIA	3		EQU	0E4h
	= 00E5	OP_E5_READ_LONG			EQU	0E5h
	= 00E6	OP_E6_WRITE_LO	NG		EQU	0E6h
210						
211						
212		;				
213			ller I/O ports			:
214 215		;				
	= 0320	IO PORT 320 DA	ra.		EOU	320h
	= 0320		AD_STATUS_WRITE	DECET		321h
	= 0322		AD_CONFIG_WRITE			322h
	= 0323	IO_PORT_323_DM		_022201	EQU	323h
220	0020				-20	3201
221						
222		;				
223		; DMA mod	de register			:
224		;				:
225			L=3 address i	ncrement		:
226						
		; demai	nd mode read/	write		:
227		; demai	nd mode read/	write		:
227 228		;	nd mode read/	write		
227 228 229	= 0047	;DRQ3_READ	nd mode read/	write	EQU	47H
227 228 229 230	= 0047 = 004B	;	nd mode read/	write	EQU EQU	
227 228 229 230 231		;DRQ3_READ	nd mode read/	write		47H
227 228 229 230 231 232		;DRQ3_READ	nd mode read/	write		47H
227 228 229 230 231 232 233		DRQ3_READ DRQ3_WRITE		write		47H 4BH
227 228 229 230 231 232 233 234		DRQ3_READ DRQ3_WRITE	nd mode read/	write		47H
227 228 229 230 231 232 233		DRQ3_READ DRQ3_WRITE		write		47H 4BH
227 228 229 230 231 232 233 234 235		DRQ3_READ DRQ3_WRITE		write		47H 4BH
227 228 229 230 231 232 233 234 235 236	= 004B	; DRQ3_READ DRQ3_WRITE ; ; Main R(;	OM image			47H 4BH
227 228 229 230 231 232 233 234 235 236 237	= 004B	; DRQ3_READ DRQ3_WRITE ; ; Main R(;	OM image			47H 4BH
227 228 229 230 231 232 233 234 235 236 237 238	= 004B	; DRQ3_READ DRQ3_WRITE ; ; Main R(;	OM image			47H 4BH
227 228 229 230 231 232 233 234 235 236 237 238 239 240 241	= 004B	DRQ3_READ DRQ3_WRITE ; ; Main R(;	OM image SEGMENT ASSUME cs:RO			47H 4BH
227 228 229 230 231 232 233 234 235 236 237 238 239 240 241	= 004B 0000 0000 AA55 0002 10	; DRQ3_READ DRQ3_WRITE ; Main R(OM image SEGMENT ASSUME cs:ROI DW 0AA55: DB 16	M b	EQU	47H 4BH
227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243	= 004B 0000 0000 AA55	; DRQ3_READ DRQ3_WRITE ; Main R(OM image SEGMENT ASSUME cs:ROI DW 0AA55: DB 16		EQU	47H 4BH
227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244	= 004B 0000 0000 AA55 0002 10 0003 E9 0A17 R	; DRQ3_READ DRQ3_WRITE ; Main R(OM image SEGMENT ASSUME cs:RO DW 0AA55: DB 16 jmp near	M b	EQU	47H 4BH
227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245	= 004B 0000 0000 AA55 0002 10 0003 E9 0A17 R 0015	; DRQ3_READ DRQ3_WRITE ; Main R(SEGMENT ASSUME cs:ROI DW 0AA55: DB 16 jmp near ; ORG 015h	M h ptr ROM_IN	EQU	47H 4BH
227 228 229 230 231 232 233 234 235 236 237 238 240 241 242 243 244 245 246	= 004B 0000 0000 AA55 0002 10 0003 E9 0A17 R 0015 0015 EB 19	; DRQ3_READ DRQ3_WRITE ; Main R(SEGMENT ASSUME cs:RO DW 0AA55 DB 16 jmp near j ORG 015h jmp short	M b	EQU	47H 4BH
227 228 229 230 231 232 233 234 235 236 237 238 240 241 242 243 244 245 246 247	= 004B 0000 0000 AA55 0002 10 0003 E9 0A17 R 0015	; DRQ3_READ DRQ3_WRITE ; Main R(SEGMENT ASSUME cs:ROI DW 0AA55: DB 16 jmp near ; ORG 015h	M h ptr ROM_IN	EQU	47H 4BH
227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248	= 004B 0000 0000 AA55 0002 10 0003 E9 0A17 R 0015 0015 EB 19 0017 90	; DRQ3_READ DRQ3_WRITE ; Main R(SEGMENT ASSUME cs:ROM DW 0AA55: DB 16 jmp near 1 ORG 015h jmp short	M h ptr ROM_IN	EQU	47H 4BH
227 228 229 230 231 232 233 234 235 236 237 238 240 241 242 242 242 242 242 244 245 244 245 244	= 004B 0000 0000 AA55 0002 10 0003 E9 0A17 R 0015 0015 EB 19 0017 90 0020	DRQ3_READ DRQ3_WRITE ; Main Re ; Main Re ROM MAGIC LENGTH	SEGMENT ASSUME cs:RO DW 0AA55 DB 16 jmp near j ORG 015h jmp short	M h ptr ROM_IN	EQU	47H 4BH
227 228 229 230 231 232 233 234 235 236 237 240 241 242 243 244 245 246 247 248 249 250	= 004B 0000 0000 AA55 0002 10 0003 E9 OA17 R 0015 0015 EB 19 0017 90 0020 0020	; DRQ3_READ DRQ3_WRITE ; Main R(SEGMENT ASSUME cs:ROI DW OAA55: DB 16 jmp near j ORG 015h jmp short nop ORG 020h	M h ptr ROM_IN	EQU	47H 4BH
227 228 229 230 231 232 233 234 235 237 240 241 242 242 243 244 245 247 248 249 249 251	= 004B 0000 0000 AA55 0002 10 0003 E9 OA17 R 0015 EB 19 0017 90 0020 0020 0020 0020 0020 00267	DRQ3_READ DRQ3_WRITE ; Main Re ; Main Re ROM MAGIC LENGTH	SEGMENT ASSUME cs:ROI DW 0AA55 DB 16 jmp near 1 ORG 015h jmp short nop ORG 020h MAX_CYLS	M h ptr ROM_IN FORMATTER	EQU	47H 4BH
227 228 229 230 231 232 233 234 235 236 237 238 240 241 242 243 244 245 246 247 248 249 250 251 252	= 004B 0000 0000 AA55 0002 10 0003 E9 0A17 R 0015 EB 19 0017 90 0020 0020 0020 0020 0267 0022 04	DRQ3_READ DRQ3_WRITE ; Main Re ; Main Re ROM MAGIC LENGTH	SEGMENT ASSUME CS:ROI DW 0AA55 DB 16 jmp near; ORG 015h jmp short nop ORG 020h MAX_CYLS MAX_HEADS	M h ptr ROM_IN	EQU	47H 4BH : : : ; ROM length in 16-byte paragraphs
227 228 229 230 231 232 233 234 235 236 237 238 240 241 242 243 244 245 246 247 248 249 250 251 252	= 004B 0000 0000 AA55 0002 10 0003 E9 OA17 R 0015 EB 19 0017 90 0020 0020 0020 0020 0020 00267	DRQ3_READ DRQ3_WRITE ; Main Re ; Main Re ROM MAGIC LENGTH	SEGMENT ASSUME cs:ROI DW 0AA55 DB 16 jmp near 1 ORG 015h jmp short nop ORG 020h MAX_CYLS	M h formatter DW DB	EQU	47H 4BH
227 228 229 230 231 233 234 235 236 237 240 241 242 243 244 245 246 247 248 249 250 251 252	= 004B 0000 0000 AA55 0002 10 0003 E9 0A17 R 0015 EB 19 0017 90 0020 0020 0020 0020 0267 0022 04	DRQ3_READ DRQ3_WRITE ; Main Re ; Main Re ROM MAGIC LENGTH	SEGMENT ASSUME CS:ROI DW 0AA55 DB 16 jmp near; ORG 015h jmp short nop ORG 020h MAX_CYLS MAX_HEADS	M h formatter DW DB	EQU	47H 4BH ; ROM length in 16-byte paragraphs ; Cylinder up to which reduced
227 228 229 230 231 232 233 234 235 236 237 240 241 245 242 243 244 245 250 251 252 253 254 255 255	= 004B 0000 0000 AA55 0002 10 0003 E9 OA17 R 0015 EB 19 0017 90 0020 0020 0020 0020 0020 0267 0022 04 0023 0064 0025 015E	DRQ3_READ DRQ3_WRITE ; Main Re ; Main Re ROM MAGIC LENGTH	SEGMENT ASSUME CS:ROI DW OAA55: DB 16 jmp near; ORG 015h jmp short nop ORG 020h MAX_CYLS MAX_HEADS RWC WPC	M TOTT ROM_IN FORMATTER DW DB DW DW	615 4 100 350	47H 4BH ; ROM length in 16-byte paragraphs ; Cylinder up to which reduced ; write current is used
227 228 229 230 231 232 233 234 235 240 241 242 243 244 245 245 246 251 252 253 253 254 255 255 255	= 004B 0000 0000 AA55 0002 10 0003 E9 0A17 R 0015 EB 19 0017 90 0020 0020 0020 0020 0020 00267 0022 04 0023 0064 0025 015E 0027 07	DRQ3_READ DRQ3_WRITE ; Main Re ; Main Re ROM MAGIC LENGTH	SEGMENT ASSUME cs:ROM DW 0AA55: DB 16 jmp near j ORG 015h jmp short nop ORG 020h MAX_CYLS MAX_HEADS RWC WPC ECC_LEN	M h formatter DW DB DW DW DW	615 4 100 350	47H 4BH ; ROM length in 16-byte paragraphs ; Cylinder up to which reduced ; write current is used ; Cylinder from which write
227 228 229 230 231 232 233 234 235 236 237 242 242 242 243 244 245 246 247 250 251 252 253 254 255 255 255 255 255 255 257	= 004B 0000 0000 AA55 0002 10 0003 E9 OA17 R 0015 EB 19 0017 90 0020 0020 0020 0267 0022 04 0023 0064 0025 015E 0027 07 0028 01	DRQ3_READ DRQ3_WRITE ; Main Re ; Main Re ROM MAGIC LENGTH	SEGMENT ASSUME cs:ROI DW 0AA55: DB 16 jmp near; ORG 015h jmp short nop ORG 020h MAX_CYLS MAX_HEADS RWC WPC ECC_LEN OPT_FLAGS	M h formatter DW DB DW DB DB	615 4 100 350	47H 4BH ; ROM length in 16-byte paragraphs ; Cylinder up to which reduced ; write current is used ; Cylinder from which write
227 228 229 230 231 232 233 234 235 236 237 240 241 242 243 245 246 247 248 249 251 252 253 254 255 254 255 256 257 258	= 004B 0000 0000 AA55 0002 10 0003 E9 OA17 R 0015 EB 19 0017 90 0020 0020 0020 0020 0020 0267 0022 04 0023 0064 0025 015E 0027 07 0028 01 0029 04	DRQ3_READ DRQ3_WRITE ; Main Re ; Main Re ROM MAGIC LENGTH	SEGMENT ASSUME CS:RON DW OAA55: DB 16 jmp near ; ORG 015h jmp short nop ORG 020h MAX_CYLS MAX_HEADS RWC WPC ECC_LEN OPT_FLAGS TIMEOUT_STD	M h petr ROM_IN FORMATTER DW DB DW DW DB DB DB DB	615 4 100 350 7 1	47H 4BH ; ROM length in 16-byte paragraphs ; Cylinder up to which reduced ; write current is used ; Cylinder from which write
227 228 239 230 231 232 233 234 235 236 240 241 242 243 244 245 255 255 255 256 257 258 259 269	= 004B 0000 0000 AA55 0002 10 0003 E9 OA17 R 0015 EB 19 0017 90 0020 0020 0020 0020 0267 0022 04 0023 0064 0025 015E 0027 07 0028 01 0029 04 0023 09	DRQ3_READ DRQ3_WRITE ; Main Re ; Main Re ROM MAGIC LENGTH	SEGMENT ASSUME CS:ROW DW OAA55: DB 16 jmp near; ORG 015h jmp short nop ORG 020h MAX_CYLS MAX_HEADS RWC WPC ECC_LEN OPT_FLAGS TIMEOUT_STD TIMEOUT_FMT	DW DB	615 4 100 350 7 1 4	47H 4BH ; ROM length in 16-byte paragraphs ; Cylinder up to which reduced ; write current is used ; Cylinder from which write
227 228 229 230 231 232 233 234 235 236 237 240 241 242 243 244 245 250 251 255 255 256 257 258 259 260 261	= 004B 0000 0000 AA55 0002 10 0003 E9 0A17 R 0015 EB 19 0017 90 0020 0020 0020 0267 0022 04 0023 0064 0025 015E 0027 07 0028 01 0029 04 0020 02 0020 02 0020 02 0020 02 0020 02 0020 02 0020 02 0020 02 0020 02 0020 02 0020 02 0020 02 0020 02 0020 02 0020 02 0020 02 0020 02	DRQ3_READ DRQ3_WRITE ; Main Re ; Main Re ROM MAGIC LENGTH	SEGMENT ASSUME CS:ROI DW OAA55: DB 16 jmp near ; ORG 015h jmp short nop ORG 020h MAX_CYLS MAX_HEADS RWC WPC ECC_LEN OPT_FLAGS TIMEOUT_STD TIMEOUT_CHK	DW DB	615 4 100 350 7 1 4 9	47H 4BH ; ROM length in 16-byte paragraphs ; Cylinder up to which reduced ; write current is used ; Cylinder from which write
227 228 229 230 231 232 233 234 235 236 237 240 241 242 243 244 245 250 251 255 255 256 257 258 259 260 261	= 004B 0000 0000 AA55 0002 10 0003 E9 OA17 R 0015 EB 19 0017 90 0020 0020 0020 0020 0267 0022 04 0023 0064 0025 015E 0027 07 0028 01 0029 04 0023 09	DRQ3_READ DRQ3_WRITE ; Main Re ; Main Re ROM MAGIC LENGTH	SEGMENT ASSUME CS:ROW DW OAA55: DB 16 jmp near; ORG 015h jmp short nop ORG 020h MAX_CYLS MAX_HEADS RWC WPC ECC_LEN OPT_FLAGS TIMEOUT_STD TIMEOUT_FMT	DW DB	615 4 100 350 7 1 4	47H 4BH ; ROM length in 16-byte paragraphs ; Cylinder up to which reduced ; write current is used ; Cylinder from which write

```
PAGE
265
266
267
                                                     Formatter utility
268
                                                     Run it from PC DOS:
269
270
                                                     >DEBUG
-G=C800:15
271
                                                     Note the disasembly of the formatter has
not been paid much attention to and the
quality probably reflects that.
272
273
274
275
                                          FORMATTER
277
      0030
                                                                          PROC NEAR
                                                                push
278
      0030 1E
279
      0031 07
                                                               pop
280
      0032 OE
                                                                push
                                                                          CS
281
      0033 1F
                                                                pop
ASSUME
283
      0034 BA 07BC R
                                                                          dx, offset STRING FORMAT WILL DESTROY
                                                               mov
      0037 E8 0755 R
003A E8 04D5 R
003D 73 03
                                                                          PRINT CRLF STRING
284
                                                                call
285
286
                                                                          ASK_CONFIRMATION
short READ_DRIVE_NUMBER
                                                                call
                                                                jnc
287
      003F E9 00F3 R
                                                                jmp
                                                                          ERROR_EXIT
288
289
      0042
                                          READ_DRIVE_NUMBER:
      0042 BA 07F4 R
0045 E8 0755 R
0048 26: C6 06 01BB R 01
004E 26: C6 06 01BA R 01
290
                                                                          dx, offset STRING_ENTER_DRIVE
                                                                          PRINT_CRLF_STRING
es:READ_NUMBER_MAX_DIGITS, 1
291
                                                               call
292
                                                               mov
293
                                                               mov
                                                                           es:READ_NUMBER_MIN_DIGITS, 1
      0054 E8 0527 R
0057 26: A1 00B8 R
005B 3C 00
294
                                                                call
                                                                          ax, es:READ_NUMBER_RESULT
                                                                           READ_NUMBER
                                                               mov
296
                                                                cmp
297
      005D 74 09
005F 3C 01
0061 74 05
                                                                je
cmp
                                                                           short GOT_VALID_DRIVE_NUMBER
298
299
                                                                           short GOT_VALID_DRIVE_NUMBER
                                                               je
call
300
      0063 E8 0765 R
                                                                          DO BEEP
      0066 EB DA
                                                                           short READ_DRIVE_NUMBER
                                                               jmp
302
303
      0068
                                          GOT_VALID_DRIVE_NUMBER:
      0068 26: A2 01B9 R
006C 3C 00
                                                                          es:HARD_DISK_NUMBER, al
305
                                                                cmp
                                                                          al, 0
                                                                           short $+2
306
      006E 75 00
                                                                           short A_BIT_FORWARD
307
308
      0070 EB 01
0072 90
                                                                jmp
                                                                                                                    ; Why...
      0073 E8 03DA R
309
                                          A BIT FORWARD:
                                                                call
                                                                          READ DRIVE PARAMS AND MORE
310
311
      0076 EB 01
0078 90
                                                                           short YET_FURTHER
                                                                jmp
                                                                пор
312
      0079 E8 00F8 R
                                          YET_FURTHER:
                                                                call
                                                                          READ BAD BLOCKS TABLE
313
314
                                                                          short GOOD_BAD_BLOCKS_TABLE
short ERROR_EXIT
      007C 73 03
                                                                jmp
315
      0080 90
                                                                пор
316
317
318
      0081
0081 BF 0700 R
                                          GOOD_BAD_BLOCKS_TABLE:
                                                                          di. offset SECTOR BUFFER
                                                               mov
                                                                          bx, di
dx, 80h
al, es:HARD_DISK_NUMBER
319
      0084 8B DF
                                                               mov
      0086 BA 0080
0089 26: A0 01B9 R
                                                               mov
321
                                                               mov
      008D 0A D0
008F B9 0066
0092 FC
                                                                          dl, al
cx, 102
322
                                                               or
323
324
                                                                                                          ; 510 / 5
                                                                cld
325
326
327
      0093
0093 B8 31C6
                                          FILL_SECTOR_BUFFER:
                                                                                                          ; First 510 bytes
                                                                          ax, 31C6h
                                                               mov
      0096 26: 89 05
0099 83 C7 02
009C B8 638C
009F 26: 89 05
00A2 83 C7 02
00A5 B0 18
328
                                                                mov
                                                                          es:[di], ax
329
330
                                                                add
                                                                          di, 2
ax, 638Ch
                                                               mov
331
                                                               mov
                                                                          es:[di], ax
                                                                          di, 2
al, 18h
                                                                add
      00A7 26: 88 05
334
                                                               mov
                                                                          es:[di], al
335
      0088 47
                                                                inc
loop
      00AB E2 E6
                                                                          FILL_SECTOR_BUFFER
                                                                                                          ; Last 2 bytes to 512
337
      00AD B8 31C6
                                                               mov
                                                                          ax, 31C6h
                                                                          es:[di], ax
338
      00B0 26: 89 05
      00B3 B8 0F01
                                                                          ax, 0F01h
340
                                                               mov
                                                                                                          ; AH=OF Write sector buffer, AL=01 One sector
      00B6 B9 0001
00B9 CD 13
00BB 80 FC 00
                                                                          cx, 1
13h
341
                                                                mov
342
343
                                                                int
                                                                          ah, 0
                                                                cmp
344
      00BE 74 09
                                                                           short WRITE_SUCCEEDED
      00C0 BA 09A0 R
00C3 E8 0707 R
                                                                          dx, offset STRING_BEEP_SECTOR_WRITE_FAIL
PRINT_MESSAGE_AND_DUMP_CONTROLLER_BUFFER
short PROCEED_REBOOTING
345
346
                                                                call
347
      00C6 EB 1C
348
      00C8 90
349
350
      00C9
                                          WRITE SUCCEEDED:
      00C9 E8 04F6 R
00CC BA 091F R
351
                                                                call
                                                                           READ_INTERLEAVE
                                                                          dx, offset STRING_CONFIRM_FORMAT
PRINT_CRLF_STRING
                                                                mov
call
353
      00CF E8 0755 R
354
355
      00D2 E8 0210 R
00D5 73 03
                                                                call
jnc
                                                                          READ_YES_OR_NO
short START_FORMATTING
356
      00D7 EB 1A
                                                                jmp
                                                                          short ERROR_EXIT
357
      00D9 90
      OODA
359
                                          START_FORMATTING:
360
      00DA E8 0425 R
                                                                call
                                                                          DO FORMAT
361
362
      00DD 72 05
00DF E8 030F R
                                                                jc
call
                                                                          short PROCEED_REBOOTING
DO_FORMAT_BAD
363
      00E2 72 00
                                                                           short $+2
364
365
      00E4
00E4 BA 0981 R
                                          PROCEED_REBOOTIN
                                                                           dx, offset STRING_ANY_KEY_TO_REBOOT
366
      00E7 E8 0755 R
                                                               call
                                                                          PRINT_CRLF_STRING
367
368
      00EA B4 08
00EC CD 21
                                                                          ah, 8
21h
                                                                mov
                                                                                                          ; Keyboard input
369
      OOEE EA FFFO ---- R
                                                               jmp
                                                                          RESET VEC
370
      00F3
                                          ERROR_EXIT:
                                                                           ax, 4C00h
                                                                          21h
372
      00F6 CD 21
                                                               int
373
374
      00F8
                                          FORMATTER
                                                                          ENDP
375
```

```
PAGE
      OOFS
                                            READ_BAD_BLOCKS_TABLE
                                                                             PROC NEAR
378
379
      00F8 26: C6 06 01EA R 00
00FE 26: C6 06 01EB R 00
                                                                              es:BAD_BLOCKS_TABLE_USED_SPACE, 0
es:NUM_BAD_BLOCK_ENTRIES, 0
                                                                  mov
      0104 BA 08B5 R
0107 E8 0755 R
010A E8 0210 R
380
                                                                              dx, offset STRING_ANY_DEFECTS
                                                                              PRINT_CRLF_STRING
382
                                                                  call
                                                                             READ_YES_OR_NO
short DO READ DEFECTS
383
      010D 73 03
                                                                  jnc
384
385
      010F E9 01BB R
                                                                             DEFECTS_READ_DONE
                                                                   jmp
      0112
386
                                           DO_READ_DEFECTS:
      0112 BA 08CA R
0115 E8 0755 R
                                                                             dx, offset STRING_PRESS_RET_TO_END
PRINT_CRLF_STRING
389
390
      0118
                                           READ_CYLINDER:
392
      0118 BA 08EC R
                                                                              dx, offset STRING_CYLINDER
                                                                             PRINT_CRLF_STRING
MOVE_CURSOR
es:READ_NUMBER_MAX_DIGITS, 4
      011B E8 0755 R
011E E8 0795 R
                                                                  call
393
      0121 26: C6 06 01BB R 04
0127 26: C6 06 01BA R 02
395
                                                                  mov
396
                                                                  mov
                                                                              es:READ_NUMBER_MIN_DIGITS, 2
397
398
      012D E8 0527 R
0130 26: 80 3E 00BA R 00
                                                                             READ_NUMBER
es:READ_NUMBER_RESULT_LEN, 0 ; Empty input?
                                                                  call
                                                                  jmp
399
      0136 75 03
                                                                              short CYLINDER WAS READ
400
401
      0138 EB 6D
013A 90
                                                                              short ENTRY_READ_DONE
                                                                   nop
402
403
      013B
                                           CYLINDER_WAS_READ:
      013B 26: A1 00B8 R
013F 26: 39 06 01C8 R
0144 77 05
0146 E8 0780 R
                                                                             ax, es:READ_NUMBER_RESULT
404
                                                                  mov
                                                                  cmp
ja
405
                                                                             es: LAST CYLINDER NUMBER. ax
                                                                             short CYLINDER_CHECK_SUCCESSFUL
BAD_ENTRY_BEEP
short READ_CYLINDER
406
407
                                                                   call
408
      0149 EB CD
                                                                  qmj
409
410
411
      014B
                                           CYLINDER_CHECK_SUCCESSFUL:
412
      014B 26: A3 01DA R
                                                                              es:CYLINDER_NUMBER, ax
413
414
      014F
                                           READ_HEAD:
      014F BA 08FB R
415
                                                                  mov
                                                                              dx. offset STRING HEAD
416
      0152 E8 0755 R
0155 E8 0795 R
                                                                  call
                                                                             PRINT_CRLF_STRING
MOVE_CURSOR
417
                                                                  call
      0155 E8 075 R

0158 26: C6 06 01BB R 02

015E 26: C6 06 01BA R 01

0164 E8 0527 R

0167 26: A1 00B8 R

016B 26: 38 06 01CC R

0170 76 19
418
                                                                  mov
                                                                              es:READ NUMBER MAX DIGITS. 2
419
420
                                                                  mov
call
                                                                             es:READ_NUMBER_MIN_DIGITS, 1
READ_NUMBER
                                                                             READ_NUMBER
ax, es:READ_NUMBER_RESULT
es:LAST_HEAD_NUMBER, al
short BAD_HEAD_READ_AGAIN
421
                                                                  mov
422
423
                                                                  cmp
      0172 26: 8B 0E 01DA R
424
                                                                   mov
                                                                              cx, es:CYLINDER_NUMBER
425
      0177 41
                                                                   inc
                                                                              CX
CX, es:LAST_CYLINDER_NUMBER
      0178 26: 3B 0E 01C8 R
017D 72 11
                                                                   cmp
                                                                  'nЬ
427
                                                                              short HEAD READ SUCCESSFUL
      0175 72 11
017F 26: 8A 26 01CC R
0184 2A E0
0186 80 FC 01
                                                                             ah, es:LAST_HEAD_NUMBER
ah, al
ah, 1
428
                                                                  mov
sub
430
                                                                   cmp
431
      0189 77 05
                                                                              short HEAD_READ_SUCCESSFUL
      018B
                                           BAD_HEAD_READ_AGAIN:
433
      018B E8 0780 R
018E EB BF
434
                                                                  call.
                                                                             BAD_ENTRY_BEEP
435
436
                                                                              short READ_HEAD
                                                                  jmp
437
      0190
                                            HEAD_READ_SUCCESSFUL:
      0190
0190 26: A2 01DC R
0194 E8 01CC R
0197 26: FE 06 01EB R
019C 26: 80 3E 01EB R C0
01A2 73 11
438
439
                                                                              es:HEAD NUMBER, al
                                                                   call
                                                                              PROCESS_BAD_BLOCKS_ENTRY
440
                                                                  inc
                                                                              es: NUM BAD BLOCK ENTRIES
441
442
                                                                  cmp
                                                                              es:NUM_BAD_BLOCK_ENTRIES, 0C0h
short DEFECTS_TABLE_FULL
      01A4 E9 0118 R
443
                                                                   jmp
                                                                             READ_CYLINDER
444
      01A7
                                            ENTRY_READ_DONE:
      01A7 BA 090A R
                                                                              dx, offset STRING_ASK_MORE_DEFECT_ENTRIES
446
                                                                  mov
                                                                             PRINT_CRLF_STRING
READ_YES_OR_NO
short DEFECTS_READ_DONE
447
      01AA E8 0755 R
01AD E8 0210 R
                                                                  call
449
      01B0 72 09
                                                                  jç
      01B2 E9 0118 R
450
                                                                   jmp
                                                                              READ_CYLINDER
      01B5
                                           DEFECTS_TABLE_FULL:
452
      01B5 BA 089B R
01B8 E8 0755 R
                                                                             dx, offset STRING_NO_MORE_DEFECTS
PRINT_CRLF_STRING
453
454
455
456
      01BB
                                           DEFECTS_READ_DONE:
      01BB 26: 80 3E 01EB R 00
01C1 74 05
01C3 E8 04D5 R
457
                                                                              es:NUM_BAD_BLOCK_ENTRIES, 0
458
                                                                              short SUCCESS
                                                                  je
459
                                                                  call
                                                                             ASK CONFIRMATION
460
      01C6 72 02
                                                                              short ABORTED
461
462
      01C8
                                           SUCCESS:
      01C8 F8
01C9 C3
463
                                                                  clc
                                                                  ret
                                           ABORTED:
465
      01CA
      01CA F9
466
                                                                  stc
                                           READ_BAD_BLOCKS_TABLE
                                                                            ENDP
468
      01CC
469
```

```
PAGE
472
       01CC
                                                   PROCESS_BAD_BLOCKS_ENTRY
                                                                                                       PROC NEAR
       O1CC
O1CC BB 01E6 R
O1CF 26: A1 01DA R
O1D3 26: 88 47 03
O1D7 80 E4 07
O1DA B1 06
473
474
                                                                                          bx, offset CYLINDER_TABLE ax, es:CYLINDER_NUMBER
                                                                             mov
                                                                                          es:[bx+3], al
ah, 7
cl, 6
475
476
477
                                                                              and
                                                                              sh1
                                                                                          ah, cl
short HDD
es:HEAD_NUMBER, 80h
478
        01DC D2 E4
479
480
        01DE 73 06
01E0 26: 80 0E 01DC R 80
                                                                             or
481
        01E6
                                                   HDD:
       01E6
01E6 26: 88 67 02
01EA 26: A0 01DC R
01EE 26: 88 47 01
01F2 E8 0451 R
01F5 BB 0400 R
                                                                                          es:[bx+2], ah
al, es:HEAD_NUMBER
es:[bx+1], al
FORMAT_BAD_BLOCK_ENTRY
482
483
                                                                             mov
484
                                                                             mov
                                                                             call
mov
485
                                                                                           bx, offset BAD_BLOCKS_TABLE
        01F8 33 C9
487
                                                                             xor
                                                                                          cx, cx
       01F8 33 C9

01F8 26: 8A 0E 01EA R

01FF 03 D9

0201 26: 89 07

0204 26: 88 57 02

0208 26: 80 06 01EA R 04

020E F8
                                                                                          cx, cx
cl, es:BAD_BLOCKS_TABLE_USED_SPACE
bx, cx
es:[bx], ax
es:[bx], dl
es:BAD_BLOCKS_TABLE_USED_SPACE, 4
488
489
                                                                              mov
add
490
                                                                             mov
491
492
493
                                                                              add
                                                                              clc
494
       020F C3
                                                                              ret
495
496
497
        0210
                                                   PROCESS_BAD_BLOCKS_ENTRY
                                                                                                       ENDP
498
499
        0210
                                                   READ_YES_OR_NO
                                                                                          PROC NEAR
       0210 E8 06D3 R
0213
0213 B4 01
0215 CD 21
500
                                                                             call
                                                                                          READ_CURSOR_POS
501
502
                                                   READ_ANSWER:
                                                                                                                                 ; Read keyboard
                                                                                          21h
503
                                                                              int
504
505
506
       0217 50
0218 B4 08
021A CD 21
                                                                             push
mov
                                                                                                                                 ; Read, no echo
                                                                              int
                                                                                          21h
       021C 3C 0D
021E 74 09
0220 58
                                                                              cmp
je
                                                                                          al, ODh
507
                                                                                                                                 ; <ENTER>
508
509
                                                                                           short CONFIRMED_ENTER
                                                                             pop
                                                                                          ax
510
        0221
                                                   BAD ANSWER:
       0221
0221 E8 0765 R
0224 E8 06EE R
0227 EB EA
                                                                                          DO_BEEP
SET_CURSOR_POS
512
513
                                                                              call
                                                                              jmp
                                                                                          short READ_ANSWER
514
515
       0229
0229 58
                                                   CONFIRMED_ENTER
                                                                             pop
                                                                                          al, 5Fh
al, 'Y'
short ANSWER_NO
516
        022A 24 5F
                                                                              and
                                                                             cmp
jne
clc
517
518
       022C 3C 59
022E 75 02
519
        0230 F8
       0231 C3
0232
0232 3C 4E
520
521
                                                   ANSWER_NO:
                                                                                          al, 'N'
522
                                                                             cmp
523
524
525
       0234 75 EB
0236 F9
                                                                             jne
stc
                                                                                          short BAD_ANSWER
        0237 C3
                                                                             ret
526
        0238
                                                   READ_YES_OR_NO
                                                                                          ENDP
527
528
529
530
531
       0238
0238 26: A3 01E3 R
                                                   STORE_CURRENT_BAD_ENTRY PROC NEAR mov es:CURRENT_BAD_ENTRY, ax
                                                                             mov
       023C 26: 88 16 01E5 R
0241 C3
0242
532
                                                                                          es:CURRENT_BAD_ENTRY_0, dl
533
534
                                                   ret
STORE_CURRENT_BAD_ENTRY ENDP
535
536
537
                                                   LOAD_CURRENT_BAD_ENTRY PROC NEAR
538
       0242
       0242 33 D2
0244 26: A1 01E3 R
0248 26: 8A 16 01E5 R
                                                                                          dx, dx
ax, es:CURRENT_BAD_ENTRY
dl, es:CURRENT_BAD_ENTRY_0
541
                                                                             mov
       024D C3
                                                   ret
LOAD_CURRENT_BAD_ENTRY ENDP
544
545
```

```
PAGE
547
548
549
                                                       The Formatter Victor V86P Hard Drive ROM has verification neutered. The bad blocks
550
                                                       had to be entered manually.
                                                      The disassembly quality of these unreachable routines is thus even lower. Sorry.
552
553
554
555
556
      024E
                                           UNUSED_VERIFY
                                                                             PROC NEAR
      024E BA 087B R
0251 E8 0755 R
0254 26: C7 06 01FA R 0000
025B E8 036D R
                                                                             dx, offset STRING_VERIFYING
PRINT_CRLF_STRING
es:UNK_VERIFY_UNUSED, 0
557
                                                                  call
559
                                                                      mov
                                                                  call.
560
                                                                            UNK_VERIFY_2
                                                                  xor
                                                                             dx, dx
562
      0260 33 CO
                                                                  xor
                                                                             ax, ax
563
      0262
                                            VERIFY_NEXT:
      0262 50
565
                                                                  push
                                                                             ax
                                                                  push
566
      0263 52
                                                                             dx
567
568
      0264 E8 0496 R
                                                                             CURRENT BAD ENTRY TO INT13H PARAMS
                                                                  call
                                                                                             ; Verify 17 sectors
; Verify sectors
; 512 bytes
                                                                             ax, 17
ah, 4
569
      0267 B8 0011
                                                                  mov
      026A B4 04
026C BB 0200
570
                                                                             bx, 512
571
                                                                             13h
572
      026F CD 13
                                                                  int
573
574
      0271 80 FC 00
                                                                             ah, INT13H_STATUS_00_NO_ERROR
                                                                  cmp
                                                                  je
cmp
575
      0274 74 28
                                                                             short VERIFY FOUND GOOD
      0274 74 28
0276 80 FC 11
0279 74 23
027B 80 FC 0B
                                                                             ah, INT13H_STATUS_11_ECC_FIXED
short VERIFY_FOUND_GOOD
ah, INT13H_STATUS_0B_BAD_CYLINDER
576
                                                                  je
578
                                                                  cmp
579
      027E 74 1E
                                                                             short VERIFY_FOUND_GOOD
      0280 80 FC 10
                                                                             ah, INT13H_STATUS_10_ECC_ERROR
581
                                                                  cmp
                                                                  je
cmp
                                                                             short VERIFY_FOUND_BAD
ah, INT13H_STATUS_02_ADDR_MARK_NOT_FOUND
582
      0283 74 67
583
584
      0285 80 FC 02
0288 74 62
                                                                  je
cmp
je
                                                                             short VERIFY_FOUND_BAD
                                                                             ah. INT13H STATUS_04_SECTOR_NOT_FOUND
      028A 80 FC 04
585
      028D 74 5D
028F 80 FC 28
                                                                             short VERIFY_FOUND_BAD
ah, INT13H_STATUS_28_UNKNOWN_ERROR
587
                                                                  cmp
je
588
      0292 74 58
                                                                             short VERIFY_FOUND_BAD
      0294 5A
0295 58
                                                                  pop
590
                                                                  pop
591
      0296 BA 0834 R
                                                                             dx. offset STRING VERIFY FAILED
592
593
      0299 E8 0707 R
029C F9
                                                                  call
                                                                             PRINT_MESSAGE_AND_DUMP_CONTROLLER_BUFFER
                                                                  stc
594
      029D C3
                                                                  ret
595
      029E
                                            VERIFY_FOUND_GOOD:
      029E 5A
597
                                                                  pop
                                                                             dx
      029F 58
02A0 B9 0011
02A3 03 C1
598
                                                                  pop
                                                                             аж
сж, 17
599
600
                                                                  add
                                                                             ax. cx
      02A5 80 D2 00
02A8 26: 8B 0E 01EC R
02AD 26: 8A 1E 01EE R
601
                                                                  adc
                                                                             d1, 0
                                                                             cx, es:UNK_VERIFY_STH
bl, es:UNK_VERIFY_STH_0
                                                                  mov
603
                                                                  mov
      02B2 3A DA
02B4 77 06
02B6 3B C8
                                                                  cmp
ja
604
                                                                             bl. dl
605
606
                                                                             short VERIFY_NOT_DONE_YET
                                                                  cmp
607
      02B8 77 02
                                                                  ja
                                                                             short VERIFY NOT DONE YET
608
609
       02BA EB 42
                                                                  jmp
                                                                             short VERIFY_DONE
610
      02BC
                                            VERIFY_NOT_DONE_YET:
611
612
      02BC 50
02BD 52
                                                                  push
                                                                             ax
dx
                                                                  push
                                                                             ax, es:UNK_VERIFY_UNUSED
613
      02BE 26: A1 01FA R
                                                                  mov
614
615
      02C2 B9 0011
02C5 03 C1
                                                                             cx, 11h
ax, cx
                                                                  mov
add
      02C5 03 C1
02C7 26: A3 01FA R
02CB 26: 8B 0E 01F8 R
                                                                             es: UNK_VERIFY_UNUSED, ax
616
                                                                  mov
617
                                                                  cmp
                                                                             cx, es:UNK_VERIFY_UNUSED_0
ax, cx
      02D0 3B C1
02D2 72 13
                                                                             short GO_VERIFY_NEXT
619
                                                                  jb
620
      02D4 2B C1
                                                                  sub
      02D6 26: A3 01FA R
02DA 26: A0 01EF R
                                                                             es:UNK_VERIFY_UNUSED, ax
al, byte ptr es:UNK_VERIFY_UNUSED_1
                                                                  mov
622
                                                                  mov
623
      02DE 04 05
                                                                  add
624
625
      02E0 26: A2 01EF R
02E4 E8 039E R
                                                                             byte ptr es:UNK_VERIFY_UNUSED_1, al
VERIFY_PROGRESS
                                                                  call
626
627
628
      02E7
                                            GO_VERIFY_NEXT:
      02E7 5A
                                                                             dx
                                                                  pop
629
      02E8 58
                                                                  pop
jmp
                                                                             VERIFY_NEXT
630
      02E9 E9 0262 R
631
      02EC
632
                                            VERIFY FOUND BAD:
633
      02EC 54
634
       02ED 58
                                                                  pop
call
      02EE E8 0238 R
                                                                             STORE_CURRENT_BAD_ENTRY
635
636
      02F1 E8 0350 R
02F4 73 01
                                                                  call
                                                                             FORMAT_BAD_TRACK
short BAD_FORMAT_SUCCESSFUL
                                                                  jnc
638
      02F6 C3
                                                                  ret
639
640
641
      02F7
02F7 E8 0242 R
                                            BAD_FORMAT_SUCCESSFUL:
                                                                  call
                                                                             LOAD_CURRENT_BAD_ENTRY
642
      02FA 50
                                                                  push
643
644
      02FB 52
02FC EB A0
                                                                             short VERIFY_FOUND_GOOD
                                                                  jmp
645
646
647
      02FE
02FE 26: C6 06 01EF R 64
                                            VERIFY_DONE:
                                                                             byte ptr es:UNK_VERIFY_UNUSED_1, 64h
      0304 E8 039E R
0307 BA 088A R
030A E8 0755 R
648
                                                                  call
                                                                             VERIFY PROGRESS
                                                                             dx, offset STRING_VERIFY_COMPLETE
PRINT_CRLF_STRING
                                                                  mov
call
651
      030D F8
                                                                  clc
      030E C3
      030F
                                           UNUSED_VERIFY
654
```

```
PAGE
657
      030F
                                            DO_FORMAT_BAD
                                                                              PROC NEAR
658
659
      030F 26: 80 3E 01EB R 00
0315 74 37
                                                                              es:NUM_BAD_BLOCK_ENTRIES, 0
short FORMAT_BAD_DONE
                                                                   jе
660
       0317 BA 09EC R
                                                                               dx, offset STRING_PROCESSING_DEFECTS
                                                                              PRINT_CRLF_STRING
bx, 400h
es:BAD_POINTER, 0
      031A E8 0755 R
031D BB 0400
                                                                   call
662
                                                                   mov
       0320 26: C6 06 01BE R 00
663
                                                                   mov
664
665
                                            FORMAT_TRACK:
      0326 26: A0 01BE R
032A 26: A0 01BE R
032F 74 1D
0331 26: 8B 07
0334 26: 8A 57 02
0338 E8 0238 R
666
                                                                   mov
                                                                              al, es:BAD_POINTER
                                                                   je
mov
                                                                              al, es:NUM_BAD_BLOCK_ENTRIES
short FORMAT_BAD_DONE
669
                                                                               ax, es:[bx]
                                                                              dl, es:[bx+2]
STORE_CURRENT_BAD_ENTRY
670
672
       033B 53
                                                                   push
                                                                              bx
      033C E8 0350 R
033F 73 02
0341 5B
673
                                                                   call
                                                                              FORMAT BAD TRACK
                                                                   jnc
                                                                               short NEXT_TRACK
675
                                                                   pop
ret
676
       0342 C3
677
678
       0343
                                            NEXT_TRACK:
679
       0343 5B
      0344 83 C3 04
0347 26: FE 06 01BE R
034C EB D8
680
681
                                                                               es:BAD_POINTER
                                                                   inc
682
                                                                   jmp
                                                                               short FORMAT TRACK
683
684
                                            FORMAT_BAD_DONE:
685
       034E F8
                                                                   clc
686
       034F C3
687
       0350
                                            DO_FORMAT_BAD
                                                                              ENDP
688
689
      0350
0350 E8 0242 R
                                                                              PROC NEAR
LOAD_CURRENT_BAD_ENTRY
                                             FORMAT_BAD_TRACK
691
                                                                   call
      0353 E8 0496 R
0356 B4 06
0358 26: A0 01D7 R
692
                                                                   call
                                                                               CURRENT_BAD_ENTRY_TO_INT13H_PARAMS
                                                                   mov
                                                                                                               ; Format bad track
                                                                               al, es:INTERLEAVE
694
       035C CD 13
695
                                                                   int
                                                                              13h
      035E 80 FC 00
0361 74 08
                                                                               ah, O
                                                                   стр
697
                                                                               short BAD_TRACK_FORMAT_SUCCESSFUL
                                                                   je
mov
698
       0363 BA 081C R
                                                                               dx. offset STRING BAD SECTOR FAILED
699
700
      0366 E8 0707 R
0369 F9
                                                                   call
                                                                              PRINT_MESSAGE_AND_DUMP_CONTROLLER_BUFFER
                                                                   stc
701
       036A C3
                                                                   ret
702
703
       036B
036B F8
                                            BAD_TRACK_FORMAT_SUCCESSFUL:
704
       036C C3
                                                                   ret
705
706
       036D
                                            FORMAT_BAD_TRACK
                                                                              ENDP
707
708
709
710
       036D
                                            UNK_VERIFY_2
                                                                              PROC NEAR
      036D 26: A1 01C8 R
0371 33 C9
                                                                              ax, es:LAST_CYLINDER_NUMBER
                                                                   xor
                                                                              cx. cx
711
712
      0373 26: 8A 0E 01CC R
0378 F7 E1
037A B9 0011
                                                                               cl, es:LAST_HEAD_NUMBER
                                                                   mul
                                                                              cx, 17
713
                                                                   mov
      037D F7 E1
037F B9 0014
0382 F7 F1
714
                                                                   mu1
715
716
                                                                   mov
div
      0384 26: A3 01F8 R
0388 26: C6 06 01EF R 00
038E E8 06D3 R
                                                                               es:UNK VERIFY UNUSED 0. ax
717
                                                                   mov
718
719
                                                                              byte ptr es:UNK_VERIFY_UNUSED_1, 0
READ_CURSOR_POS
                                                                   call
      0391 E8 076C R
0394 26: A2 00B2 R
0398 26: 88 26 00B3 R
                                                                              LOAD_CURSOR_POS
es:READ_NUMBER_SAVED_AL, al
es:READ_NUMBER_SAVED_AH, ah
720
                                                                   call
721
722
723
      039D C3
                                                                   ret
724
725
       039E
                                            UNK_VERIFY_2
                                                                              ENDD
726
727
728
       039E
                                            VERIFY_PROGRESS
                                                                              PROC NEAR
      039E 26: A0 00B2 R
03A2 26: 8A 26 00B3 R
03A7 E8 0776 R
                                                                              al, es:READ_NUMBER_SAVED_AL
ah, es:READ_NUMBER_SAVED_AH
729
                                                                   mov
730
                                                                   call
                                                                               STORE CURSOR POS
       03AA E8 06EE R
                                                                   call
                                                                              al, byte ptr es:UNK_VERIFY_UNUSED_1
       03AD 26: A0 01EF R
732
                                                                   mov
733
734
735
                                                                              ah, ah
word ptr es:VERIFY_STH_2, ax
dx, dx
                                                                   mov
       03B1 32 E4
      03B3 26: A3 01D6 R
03B7 33 D2
                                                                   xor
736
       03B9 E8 0626 R
                                                                   call
                                                                              VERIFY PROGRESS STH
                                                                              .Engf:_FRUGRESS_STH
si, (offset VERIFY_PROGRESS_BUFFER+1)
cx, 4
      03BC BE 01CF R
03BF B9 0004
738
739
       03C2
                                            PROGRESS ANOTHER CHAR:
740
741
742
      03C2
03C2 26: 80 3C 20
03C6 75 04
03C8 46
                                                                              byte ptr es:[si], ' '
short PROGRESS_DONE
                                                                   cmp
                                                                   jne
                                                                   inc
                                                                              si
743
744
       03C9 49
       03CA EB F6
                                                                              short PROGRESS_ANOTHER_CHAR
                                                                   jmp
                                            PROGRESS_DONE:
745
746
747
748
749
750
751
       03CC
       03CC 1E
                                                                   push
                                                                   push
       03CE 1F
                                                                   pop
                                                                   ASSUME
                                                                              ds : FORMATTER HEAP
      03CF E8 0B3C R
03D2 1F
                                                                               PRINT_CX_CHARS_FROM_DS_SI
                                                                   call
                                                                   POP
ASSUME
                                                                              ds: ZEROSEG
752
753
754
      03D3 BA 099C R
03D6 E8 074C R
                                                                   mov
call
                                                                              dx, offset STRING_PERCENT
PRINT_STRING_NO_CRLF
755
      03D9 C3
                                                                   ret
                                            VERIFY_PROGRESS
```

```
PAGE
760
761
762
                                                                                  DRE PROC NEAR
ah, 8
INT13H_FOR_CHOSEN_DRIVE
       03DA
                                               READ_DRIVE_PARAMS_AND_MORE
       03DA B4 08
03DC E8 06C5 R
                                                                                                                                ; Get drive parameters
                                                                     mov
call
763
764
765
       03DF FE C6
03E1 26: 88 36 01CC R
03E6 8A D1
                                                                      inc
                                                                                  dh
                                                                                  es:LAST_HEAD_NUMBER, dh
                                                                      mov
                                                                                  dl, cl
dl, 0C0h
       03E8 80 E2 C0
766
767
768
769
770
771
                                                                      and
       03EB B1 06
03ED D2 EA
                                                                      mov
shr
                                                                                  cl, 6
dl, cl
       03EF 8A CD
                                                                      mov
                                                                                  cl, ch
                                                                                  ch, dl
cx, 2
       03F1 8A EA
03F3 83 C1 02
                                                                      mov
add
772
773
774
                                                                      push
mov
       0386 51
       03F7 B4 FE
03F9 E8 06C5 R
03FC 59
                                                                                  ah, OFEh
                                                                                                                                 ; Undocumented disk function
                                                                                  INT13H_FOR_CHOSEN_DRIVE
775
                                                                      call
                                                                                                                                ; Perhaps this is pointless?
776
777
       03FD 03 CA
778
       03FF 26: 89 0E 01C8 R
779
                                                                      mov
                                                                                  es:LAST_CYLINDER_NUMBER, cx
780
781
       0404 8B C1
0406 48
                                                                      mov
                                                                                  ax, cx
                                                                                  ax
                                                                                  dx. dx
782
       0407 33 D2
                                                                      xor
       0409 26: 8A 0E 01CC R
040E 32 ED
                                                                                  cl, es:LAST_HEAD_NUMBER
                                                                                  ch, ch
                                                                      xor
785
       0410 F7 E1
                                                                      mul
                                                                                  CX
786
787
788
789
       0412 FE C9
0414 03 C1
                                                                      dec
add
                                                                                  cl
                                                                                  ax, cx
       0416 B9 0011
0419 F7 E1
041B 26: A3 01EC R
041F 26: 88 16 01EE R
                                                                      mov
                                                                                  cx, 11h
                                                                      mul
                                                                                  es:UNK_VERIFY_STH, ax
791
                                                                                  es:UNK_VERIFY_STH_0, dl
                                                                      mov
792
793
794
       0424 C3
0425
                                              ret
READ_DRIVE_PARAMS_AND_MORE
795
796
797
       0425
0425 26: C6 06 01EA R 00
                                              DO_FORMAT
                                                                                  PROC NEAR
                                                                                  es:BAD_BLOCKS_TABLE_USED_SPACE, 0
                                                                      mov
                                                                                  dx, offset STRING_FORMATTING PRINT_CRLF_STRING
798
       042B BA 085C R
                                                                      mov
       042E E8 0755 R
0431 B6 00
0433 B4 07
                                                                      call
800
                                                                      mov
                                                                                  dh, 0
ah, 7
                                                                                                                                 ; Format drive
801
                                                                      mov
       0433 B4 07
0435 26: A0 01D7 R
0439 E8 06C5 R
043C 80 FC 00
043F 74 08
0441 BA 080D R
802
803
                                                                      mov
call
                                                                                  al, es:INTERLEAVE
INT13H_FOR_CHOSEN_DRIVE
                                                                      cmp
je
mov
                                                                                  ah, 0
short Format_DONE
dx, offset STRING_FORMAT_FAILED
PRINT_MESSAGE_AND_DUMP_CONTROLLER_BUFFER
804
805
806
       0444 E8 0707 R
807
                                                                      call
       0447 F9
0448 C3
0449
808
809
                                                                      stc
                                              FORMAT_DONE:
810
       0449 BA 086B R
044C E8 0755 R
044F F8
811
                                                                                 dx, offset STRING_FORMAT_COMPLETE
PRINT_CRLF_STRING
812
813
                                                                      call
                                                                      clc
       0450 C3
0451
814
                                                                      ret
                                              DO_FORMAT
816
817
       818
819
                                              FORMAT_BAD_BLOCK_ENTRY
                                                                                 PROC NEAR
                                                                      push
                                                                      push
xor
820
                                                                                  cx
821
822
                                                                      mov
                                                                                  ah, es:[bx+2]
cl, 1
ah, cl
823
                                                                      mov
824
825
       045D B1 01
045F D2 EC
                                                                      mov
       0461 26: 8A 4F 01
0465 80 E1 80
0468 0A E1
046A B1 05
826
                                                                      mov
                                                                                  cl, es:[bx+1]
                                                                      and
                                                                                  cl, 5
829
                                                                      mov
       046C D2 EC
046E 33 C9
0470 26: 8A 0E 01CC R
0475 F7 E1
830
                                                                      shr
                                                                                  ah, cl
cx, cx
                                                                                  cl, es:LAST_HEAD_NUMBER
832
                                                                      mov
mul
833
       0477 26: 8A 4F 01
047B 81 E1 003F
                                                                                  cl, es:[bx+1]
cx, 3Fh
835
                                                                      and
                                                                      add
836
       047F 03 C1
837
838
       0481 B9 0011
0484 F7 E1
                                                                      mul
839
       0486 26: 8A 4F 02
                                                                      mov
                                                                                  cl, es:[bx+2]
cx, 3Fh
       048A 81 E1 003F
048E 03 C1
                                                                      add
                                                                                  ax, cx
dx, 0
       0490 83 D2 00
842
                                                                      adc
       0493 59
0494 5B
0495 C3
843
                                                                      pop
845
       0496
                                              FORMAT_BAD_BLOCK_ENTRY
                                                                                 ENDE
848
```

```
849
                                                  PAGE
850
       0496
                                                  CURRENT_BAD_ENTRY_TO_INT13H_PARAMS
                                                                                                               PROC NEAR
851
852
       0496 50
0497 52
                                                                          push
                                                                                       ax
dx
                                                                           push
       0498 33 C0
049A 26: A0 01CC R
049E B9 0011
                                                                                       ax, ax
al, es:LAST_HEAD_NUMBER
cx, 11h
853
                                                                           xor
854
855
                                                                          mov
       04A1 F7 E1
04A3 8B C8
04A5 5A
04A6 58
856
                                                                          mu1
                                                                                       CX
857
858
                                                                          pop
pop
div
push
mov
859
       04A6 58
04A7 F7 F1
04A9 50
04AA 8B C2
04AC B1 11
04AE 32 ED
860
861
                                                                                       ax, dx
cl, 11h
ch, ch
862
                                                                          mov
863
865
       04B0 80 E1 3F
                                                                           and
                                                                                       cl, 3Fh
       04B3 F6 F1
04B5 FE C4
04B7 8A CC
866
867
                                                                           div
                                                                           inc
868
                                                                          mov
mov
                                                                                       cl, ah
                                                                                       dh, al
dl, es:HARD_DISK_NUMBER
dl, 80h
869
       04B9 8A FO
870
871
       04BB 26: 8A 16 01B9 R
04C0 80 CA 80
                                                                           or
                                                                          pop
mov
and
872
       04C3 58
873
874
       04C4 8A E8
04C6 80 E4 07
                                                                                       ch, al
ah, 7
                                                                          push
mov
shl
875
       04C9 51
                                                                                       CX
       04CA B9 0006
04CD D2 E4
876
877
                                                                                       ah, cl
878
       04CF 73 00
                                                                           jnb
                                                                                       short $+2
       04D1 59
04D2 0A CC
879
                                                                                       cl, ah
       04D4 C3
881
                                                                           ret
882
883
884
       04D5
                                                  CURRENT_BAD_ENTRY_TO_INT13H_PARAMS
                                                                                                                ENDP
                                                                                       PROC NEAR dx, offset STRING_PRESS_RET_TO_PROCEED PRINT_CRLF_STRING
885
       04D5
                                                  ASK_CONFIRMATION
886
887
       04D5 BA 0953 R
04D8 E8 0755 R
                                                                          call
888
       04DB E8 06D3 R
                                                                           call.
                                                                                       READ CURSOR POS
889
890
       04DE E8 06EE R
                                                                                       SET_CURSOR_POS
                                                                          call
                                                                                       ah, 8
21h
al, 1Bh
       04E1 B4 08
891
                                                                                                                            ; Read key, no echo
       04E3 CD 21
04E5 3C 1B
04E7 75 02
892
893
                                                                           int
                                                                                                                            ; <ESC>
                                                                           cmp
                                                                          jne
stc
ret
894
                                                                                       short NOT ESC
895
896
       04E9 F9
04EA C3
897
       04EB
                                                  NOT_ESC:
       04EB 3C 0D
04ED 74 05
04EF E8 0765 R
                                                                                        al. ODh
                                                                                                                             : <ENTER>
                                                                                       short CONFIRMED_WITH_ENTER
DO_BEEP
                                                                           je
call
900
                                                                                        short READ_KEY_RESPONSE
901
902
903
       04F2 EB EA
04F4
04F4 F8
                                                                          clc
904
905
906
       04F5 C3
04F6
                                                                          ret
                                                 ASK_CONFIRMATION
907
908
909
       04F6
04F6 E8 06D3 R
                                                  READ_INTERLEAVE
                                                                                       PROC NEAR
READ_CURSOR_POS
                                                  READ_INTERLEAVE_NUMBER:
       04F9
04F9 E8 06EE R
04FC BA 09B5 R
910
                                                                                       SET_CURSOR_POS
dx, offset STRING_INTERLEAVE
PRINT_CRLF_STRING
es: READ_NUMBER_MAX_DIGITS, 2
es: READ_NUMBER_MIN_DIGITS, 1
911
912
                                                                           call
913
       04FF E8 0755 R
                                                                          call
914
915
       0502 26: C6 06 01BB R 02
0508 26: C6 06 01BA R 01
                                                                          call
                                                                                       READ_NUMBER
916
       050E E8 0527 R
       0511 26: A1 00B8 R
0515 3C 01
0517 72 09
                                                                          mov
cmp
jb
                                                                                       ax, es:READ_NUMBER_RESULT
al, 1
                                                                                       short WRONG_INTERLEAVE_NUMBER
919
       0517 72 05
0519 3C 10
051B 77 05
051D 26: A2 01D7 R
                                                                                       al, 10h
short WRONG_INTERLEAVE_NUMBER
es:INTERLEAVE, al
920
921
                                                                          cmp
ja
922
                                                                           mov
ret
923
       0521 C3
       0522
0522 E8 0765 R
925
                                                                          call
                                                                                       DO BEEP
                                                                                       short READ_INTERLEAVE_NUMBER ENDP
926
       0525 EB D2
927
928
                                                  READ_INTERLEAVE
929
```

```
PAGE
931
       0527
                                            READ_NUMBER
                                                                              PROC NEAR
932
933
      0527 E8 06D3 R
052A E8 076C R
                                                                              READ_CURSOR_POS
LOAD_CURSOR_POS
                                                                   call
                                                                   call
      052D 26: 88 26 00B3 R
0532 26: A2 00B2 R
0536 BB 00B4 R
                                                                              es:READ_NUMBER_SAVED_AH, ah
es:READ_NUMBER_SAVED_AL, al
bx, offset PARSE_NUMBER_BUFFER
cx, ||
934
                                                                   mov
936
                                                                   mov
937
       0539 B9 0000
                                                                   mov
938
939
      053C 26: C6 06 01C0 R 00
0542
                                                                               es:CURSOR_LOCATION_OR_SOMETHING, 0
                                            READ_NEXT_KEY:
940
       0542 B4 01
                                                                              ah, 1
21h
al, 0Dh
                                                                                                                                      ; Read key
                                                                   mov
int
      0544 CD 21
0546 3C 0D
                                                                   cmp
je
                                                                                                                                      ; <ENTER>
943
                                                                              short ENTER_PRESSED al, 8
944
       0548 74 3F
       054A 3C 08
946
       054C 75 10
                                                                   jne
                                                                               short NOT BACKSPACE
                                                                   call
dec
                                                                              BACKSPACE_WIPE_WITH_SPACE
es:CURSOR_LOCATION_OR_SOMETHING
       054E E8 065C R
      0551 26: FE 0E 01C0 R
0556 4B
                                                                                                                                      ; Move cursor back
949
                                                                   dec
                                                                              bx
                                                                              cx, 1
short BAD_INPUT
short READ_NEXT_KEY
       0557 83 E9 01
950
                                                                   sub
951
952
      055A 72 5F
055C EB E4
                                                                                                                                      ; Reached column -1?
                                                                   jmp
953
       055E
                                            NOT_BACKSPACE:
      055E 26: 3A 0E 01BB R 0563 74 56 0565 26: 88 07 0568 26: A0 01C0 R
                                                                              cl, es:READ_NUMBER_MAX_DIGITS short BAD_INPUT
                                                                   cmp
                                                                   je
mov
                                                                              es:[bx], al
al, es:CURSOR_LOCATION_OR_SOMETHING
956
957
958
       056C FE C0
                                                                   inc
      056E 26: A2 01CO R
0572 26: 80 3E 01BA R 03
0578 75 0B
                                                                              es:CURSOR LOCATION OR SOMETHING, al
959
                                                                   mov
                                                                              es:CURSON_INCHAITON_OR_SOMETHING, all
es:READ_NUMBER_MIN_DIGITS, 3
short NEXT_CHARACTER
es:CURSON_LOCATION_OR_SOMETHING, 1
960
                                                                   јле
       057A 26: 80 3E 01C0 R 01
962
                                                                   cmp
963
       0580 75 03
                                                                              short NEXT_CHARACTER
UPDATE_CURSOR_AND_WIPE
       0582 E8 05DC R
965
       0585
                                            NEXT_CHARACTER:
966
       0585 43
                                                                   inc
      0586 41
0587 EB B9
968
                                                                   jmp
                                                                              short READ_NEXT_KEY
969
       0589
                                            ENTER PRESSED:
      0589
0589 26: A0 01C0 R
058D 26: A2 00BA R
0591 26: 80 3E 01BA R 01
                                                                               al, es:CURSOR_LOCATION_OR_SOMETHING
                                                                              es:READ_NUMBER_RESULT_LEN, al
es:READ_NUMBER_MIN_DIGITS, 1
971
                                                                   mov
972
      0597 74 0B
0599 26: 80 3E 01C0 R 00
059F 74 18
                                                                              short ONE_DIGIT
es:CURSOR_LOCATION_OR_SOMETHING, 0
                                                                   cmp
                                                                   je
jmp
nop
975
                                                                              short ALL_GOOD
short MORE_THAN_ONE_DIGIT
      05A1 EB 0C
05A3 90
05A4
976
977
978
                                            ONE_DIGIT:
                                                                              es:CURSOR_LOCATION_OR_SOMETHING, 0 short MORE_THAN_ONE_DIGIT short BAD_INPUT
                                                                   cmp
jne
       05A4 26: 80 3E 01C0 R 00
      05AA 75 03
05AC EB 0D
981
                                                                   jmp
982
       05AE 90
                                            nop
MORE_THAN_ONE_DIGIT:
983
984
      05AF
05AF E8 05EE R
                                                                   call
                                                                              PARSE NUMBER
                                                                   jc
mov
985
       05B2 72 07
                                                                               short BAD INPUT
      05B4 26: 89 0E 00B8 R
05B9
                                                                              es:READ_NUMBER_RESULT, cx
987
                                            ALL_GOOD:
988
       05B9 F8
                                                                   clc
989
990
      05BA C3
05BB
                                            BAD_INPUT:
       05BB E8 0765 R
991
                                                                   call
                                                                              DO BEEP
      05BE 26: C6 06 01C0 R 00
05C4 26: 8A 26 00B3 R
05C9 26: AO 00B2 R
                                                                   mov
                                                                              es:CURSOR_LOCATION_OR_SOMETHING, 0
ah, es:READ_NUMBER_SAVED_AH
994
                                                                   mov
                                                                               al, es:READ NUMBER_SAVED_AL
995
996
      05CD E8 0776 R
05D0 E8 05E1 R
                                                                   call
call
                                                                              STORE_CURSOR_POS
SET_CURSOR_POS_AND_WIPE
997
      05D3 B9 0000
                                                                   mov
                                                                              cx, II
                                                                   mov
                                                                              bx, offset PARSE_NUMBER_BUFFER
READ_NEXT_KEY
      05D6 BB 00B4 R
       05D9 E9 0542 R
                                            READ_NUMBER
1000 05DC
                                                                              ENDP
1001
1002
                                            UPDATE_CURSOR_AND_WIPE PROC NEAR
1003 05DC
                                                                  call
1004 05DC E8 06D3 R
                                                                             READ_CURSOR_POS
1005 05DF EB 03
                                                                              short WIPE_THINGS_WITH_SPACES
                                            jmp shor
UPDATE_CURSOR_AND_WIPE ENDP
1006 05E1
1007
                                            SET_CURSOR_POS_AND_WIPE PROC NEAR
1009 05E1
1010 05E1 E8 06EE R
                                                                   call
                                                                              SET_CURSOR_POS
1011 05E4
                                            SET_CURSOR_POS_AND_WIPE ENDP
1012
1013
1014 05E4
                                            WIPE_THINGS_WITH_SPACES PROC NEAR
1015 05E4 BA 0948 R
1016 05E7 E8 074C R
                                                                             dx, offset STRING_SPACES
PRINT_STRING_NO_CRLF
                                                                  mov
call
1017 05EA E8 06EE R
                                                                   call
                                                                              SET_CURSOR_POS
1018 05ED C3
                                            ret
WIPE_THINGS_WITH_SPACES ENDP
1019 05EE
1020
```

```
1022
                                             PAGE
1023 05EE
                                             PARSE_NUMBER
                                                                               PROC NEAR
1024 05EE BB 00B4 R
1025 05F1 33 C9
                                                                               bx, offset PARSE_NUMBER_BUFFER cx, cx
                                                                    xor
1026 05F3 26: 80 3E 01C0 R 00
1027 05F9 74 27
1028 05FB
                                                                   cmp
je
                                                                               es:CURSOR_LOCATION_OR_SOMETHING, 0
                                                                               short LAST_DIGIT
                                             NEXT_DIGIT:
1029 05FB 26: 8A 07
                                                                               al, es:[bx]
al, '0'
                                                                   mov
1030 05FE 3C 30
1031 0600 72 22
                                                                    cmp
jb
                                                                               short BAD_NUMBER
al, '9'
1032 0602 3C 39
                                                                    cmp
                                                                               short BAD_NUMBER
a1, '0'
1033 0604 77 1E
1034 0606 2C 30
                                                                    ja
sub
1035 0608 98
                                                                    cbw
                                                                    add
dec
1036 0609 03 08
1036 0609 03 C8
1037 060B 26: FE 0E 01C0 R
1038 0610 26: 80 3E 01C0 R 00
1039 0616 74 0A
1040 0618 B8 000A
                                                                               es:CURSOR_LOCATION_OR_SOMETHING
                                                                               es:CURSOR_LOCATION_OR_SOMETHING, 0
                                                                    cmp
                                                                               short LAST_DIGIT
1041 061B F7 E1
                                                                    mul
                                                                               ÇX
                                                                               cx, ax
1042 061D 8B C8
1043 061F 43
1044 0620 EB D9
                                                                               short NEXT_DIGIT
                                                                    jmp
1045 0622
                                            LAST_DIGIT:
1045 0622 F8
1046 0622 F8
1047 0623 C3
1048 0624
                                                                    clc
                                                                   ret
                                             BAD NUMBER:
1049 0624 F9
1050 0625 C3
                                                                   ret
1051 0626
                                            PARSE NUMBER
                                                                               ENDP
1052
1053
1054 0626
                                            VERIFY_PROGRESS_STH
                                                                               PROC NEAR
1055 0626 53
1056 0627 51
1057 0628 26: C7 06 01CE R 2020
                                                                   push
push
                                                                               bx
bx
                                                                                   es:VERIFY_PROGRESS_BUFFER, 2020h
                                                                                                                                            ; Two spaces
1058 062F 26: C7 06 01D0 R 2020
1059 0636 26: C7 06 01D2 R 2020
1060 063D BB 0004
                                                                                   es:VERIFY_PROGRESS_BUFFER_0, 2020h
es:VERIFY_PROGRESS_BUFFER_1, 2020h
                                                                                                                                            ; Two spaces
; Two spaces
                                                                   mov
                                                                               bx, 4
1061 0640 B9 000A
                                                                   mov
                                                                               cx. OAb
1062 0643 26: A1 01D6 R
1063 0647
1064 0647 33 D2
                                                                               ax, word ptr es:VERIFY_STH_2
                                             ANOTHER_PROGRESS_DIGIT:
                                                                    xor
                                                                               dx, dx
1065 0649 F7 F1
1066 064B 80 C2 30
                                                                    div
                                                                               cx
dl, '0'
                                                                    add
1067 064E 26: 88 97 01CE R
                                                                   mov
                                                                               byte ptr es:VERIFY_PROGRESS_BUFFER[bx], dl
1068 0653 4B
1069 0654 3D 0000
1070 0657 75 EE
                                                                    dec
                                                                    cmp
                                                                    ine
                                                                               short ANOTHER PROGRESS DIGIT
1071 0659 59
1072 065A 5B
                                                                   pop
1073 065B C3
1074 065C
1075
                                             VERIFY_PROGRESS_STH
                                                                               ENDP
1076
1077 065C
                                             BACKSPACE_WIPE_WITH_SPACE
                                                                                         PROC NEAR
1078 065C 51
1079 065D E8 076C R
                                                                   push
call
                                                                               CX
LOAD_CURSOR_POS
                                                                   push
call
                                                                               ax
READ_CURSOR_POS
1080 0660 50
1080 0660 50
1081 0661 E8 06D3 R
1082 0664 BA 099E R
                                                                               dx, offset STRING_SPACE
                                                                   mov
1083 0667 E8 074C R
                                                                    call
                                                                               PRINT_STRING_NO_CRLF
1084 066A E8 06EE R
1085 066D 58
                                                                    call
                                                                               SET_CURSOR_POS
                                                                   pop
call
1086 066E E8 0776 R
                                                                               STORE_CURSOR_POS
1087 0671 59
1088 0672 C3
                                                                   pop
                                             BACKSPACE_WIPE_WITH_SPACE
1089 0673
                                                                                          ENDP
1090
1091
1092 0673
                                            UNK_UNUSED_1
                                                                               PROC NEAR
                                                                               ax, es:LAST_CYLINDER_NUMBER
1093 0673 26: A1 01C8 R
1094 0677 33 C9
1095 0679 26: 8A 0E 01CC R
1096 067E F7 E1
                                                                    xor
                                                                               cl, es:LAST_HEAD_NUMBER
                                                                   mov
                                                                   mul
1097 0680 B9 012C
1098 0683 F7 F1
                                                                               сж, 300
                                                                    div
                                                                               CX
1099
1100 0685 3C 00
1101 0687 77 02
                                                                    cmp
                                                                               short STILL_MORE
                                                                    ja
1102 0689 FE CO
                                                                    inc
                                                                               al
1103 068B
1104
                                             STILL_MORE:
1105 068B 26: A2 01D4 R
                                                                   mov
                                                                               es:UNK UNUSED STH. al
1105 068E 26
1106 068F F8
1107 0690 C3
1108 0691
                                                                    clc
                                                                   ret
                                             UNK UNUSED 1
                                                                               ENDP
1109
1110
1111 0691
                                                                                         PROC NEAR
                                             UNUSED_READ_CONTROL_BYTE
1112 0691 BA 09C9 R
1113 0694 E8 0755 R
                                                                   mov
call
                                                                               dx, offset STRING_CONTROL_BYTE_2
PRINT_CRLF_STRING
1114 0697 E8 06D3 R
                                                                               READ_CURSOR_POS
                                                                    call
1115 069A 26: FE 0E 00B0 R
1116 069F E8 06EE R
1117 06A2 26: C6 06 01BB R 03
1118 06A8 26: C6 06 01BA R 03
                                                                               es SAVED COLUMN
                                                                               SET_CURSOR_POS
es:READ_NUMBER_MAX_DIGITS, 3
                                                                    call
                                                                   mov
                                                                   mov
                                                                               es:READ_NUMBER_MIN_DIGITS, 3
1119 06AE E8 0527 R
1120 06B1 26: AI 00B8 R
1121 06B5 26: 80 3E 00BA R 00
                                                                   call
                                                                               READ_NUMBER
                                                                               READ_NUMBER
ax, es:READ_NUMBER_RESULT
                                                                   mov
                                                                   jne
                                                                               es: READ NUMBER RESULT LEN.
                                                                               short CONTROL_BYTE_SET a1, 2
1122 06BB 75 02
1123 06BD B0 02
                                                                                                               ; Default if unset
                                                                    mov
1124 06BF
                                             CONTROL BYTE SET:
1124 068F
1125 068F 26: A2 01D5 R
1126 06C3 F8
                                                                               es:CONTROL_BYTE, al
                                                                    clc
1127 06C4 C3
                                                                    ret
1128 06C5
                                            UNUSED_READ_CONTROL_BYTE
                                                                                          ENDP
1130
```

```
1131
                                               PAGE
1132 06C5
                                               INT13H_FOR_CHOSEN_DRIVE PROC NEAR
1133 06C5 26: 8A 16 01B9 R
1134 06CA 80 CA 80
                                                                                  dl, es:HARD_DISK_NUMBER
dl, 80h
                                                                      mov
1135 06CD B9 0001
                                                                                   cx, 1
1136 06D0 CD 13
1137 06D2 C3
                                                                       ret
1138 06D3
                                               INT13H_FOR_CHOSEN_DRIVE ENDP
1139
1140
1141 06D3
                                               READ_CURSOR_POS
                                                                                  PROC NEAR
1142 06D3 50
1143 06D4 56
                                                                      push
push
1144 06D5 53
                                                                       push
                                                                                  bx
1145 06D6 52
1146 06D7 51
                                                                       push
push
                                                                                  dx
cx
1147 06D8 B4 03
1148 06DA B7 00
1149 06DC CD 10
                                                                       mov
                                                                                  ah, 3
bh, ⊪
                                                                                                                      ; Read video cursor
                                                                       mov
1150 06DE 26: 88 36 00B1 R
                                                                                   es:SAVED_ROW, dh
                                                                       mov
                                                                                   es:SAVED_COLUMN, dl
1151 06E3 26: 88 16 00B0 R
1152 06E8 59
1153 06E9 5A
                                                                       pop
                                                                       pop
                                                                                   dx
                                                                       pop
1154 06EA 5B
1155 06EB 5E
1156 06EC 58
1157 06ED C3
                                                                       pop
                                                                                   ax
                                                                       ret
1158 06EE
                                               READ_CURSOR_POS
                                                                                   ENDE
1159
1160
1161 06EE
1162 06EE 50
1163 06EF 56
                                               SET_CURSOR_POS
                                                                                  PROC NEAR
                                                                       push
                                                                       push
                                                                                   si
1164 06F0 52
1165 06F1 53
1166 06F2 B4 02
                                                                       push
push
                                                                                   dx
bx
                                                                                   ah, 2
                                                                                                                      ; Set video cursor
                                                                       mov
1167 06F4 26: 8A 36 00B1 R
1168 06F9 26: 8A 16 00B0 R
1169 06FE B7 00
                                                                                        es:SAVED_ROW
es:SAVED_COLUMN
                                                                                  dh,
dl,
                                                                       mov
                                                                       mov
int
                                                                                   bh,
1170 0700 CD 10
                                                                                   10h
1170 0700 CB
1171 0702 5B
1172 0703 5A
1173 0704 5E
                                                                       pop
                                                                       pop
pop
                                                                                   dx
                                                                                   si
1174 0705 58
1175 0706 C3
                                                                       ret
1176 0707
                                               SET CURSOR POS
                                                                                  ENDP
1177
1178
1179 0707
                                               PRINT MESSAGE AND DUMP_CONTROLLER BUFFER
                                                                                                                      PROC NEAR
1180 0707 E8 0755 R
1181 070A 1E
1182 070B E8 1139 R
                                                                       call
push
                                                                                   PRINT_CRLF_STRING
                                                                                  ds
ZERO_DS
                                                                       call
1183
                                                                       ASSUME
                                                                                  ds: ZEROSEG
1184 070E A0 0442 R
1185 0711 E8 072E R
                                                                                  al, BDA_CONTROLLER_DATA_BUFFER_00
PRINT_HEX_BYTE
                                                                       call
1186 0714 A0 0443 R
1187 0717 E8 072E R
1188 071A A0 0444 R
                                                                                  al, BDA_CONTROLLER_DATA_BUFFER_01
PRINT_HEX_BYTE
al, BDA_CONTROLLER_DATA_BUFFER_02
                                                                       call
                                                                       mov
1189 071D E8 072E R
1190 0720 A0 0445 R
1191 0723 E8 072E R
                                                                       call
                                                                                  PRINT HEX BYTE
                                                                                  PRINT_HEX_BYTE
a1, BDA_CONTROLLER_DATA_BUFFER_03
PRINT_HEX_BYTE
                                                                       mov
call
                                                                                   dx. offset STRING CRLF_DOS
1192 0726 BA 09D9 R
                                                                       mov
1193 0729 E8 074C R
1194 072C 1F
                                                                       call
                                                                                   PRINT_STRING_NO_CRLF
                                                                       pop
1195 072D C3
1196 072E
1197
                                               PRINT_MESSAGE_AND_DUMP_CONTROLLER_BUFFER
                                                                                                                      ENDP
1198
1199 072E
1200 072E 50
1201 072F 32 FF
                                               PRINT_HEX_BYTE
                                                                                  PROC NEAR
                                                                       push
                                                                                  bh, bh
                                                                       xor
1202 0731 B0 20
1203 0733 E8 07AB R
1204 0736 58
                                                                       mov
call
                                                                       pop
                                                                                   ax
ax
1205 0737 50
1206 0738 B1 04
1207 073A D2 E8
                                                                       shr
                                                                                   al, cl
1208 073C E8 07BO R
1209 073F E8 07AB R
1210 0742 58
                                                                       call
                                                                                   NUMBER_TO_HEX_DIGIT
                                                                                   BIOS_WRITE_TEXT
                                                                       pop
and
1211 0743 24 OF
                                                                                   al. OFh
1212 0745 E8 07B0 R
1213 0748 E8 07AB R
                                                                                  NUMBER_TO_HEX_DIGIT
BIOS_WRITE_TEXT
                                                                       call
                                                                       call
1214 074B C3
1215 074C
                                                                       ret
                                               PRINT_HEX_BYTE
                                                                                   ENDP
1216
1217
1218 074C
1219 074C 1E
1220 074D 0E
                                               PRINT_STRING_NO_CRLF
                                                                                   PROC NEAR
                                                                       push
                                                                                  cs
ds
ds:ROM
                                                                       push
1221 074E 1F
1222
1223 074F B4 09
                                                                                                          ; DOS print string
                                                                       mov
                                                                                   ah, 9
21h
1224 0751 CD 21
1225 0753 1F
                                                                       int
                                                                       pop
                                                                                  ds:FORMATTER_HEAP
1226
                                                                       ASSUME
1227 0754 C3
                                                                       ret
1228 0755
1229
                                               PRINT_STRING_NO_CRLF
1230
```

1343

20 63 6F 6D 70 6C

```
1231
                                                 PAGE
 1232 0755
                                                 PRINT_CRLF_STRING
                                                                                     PROC NEAR
1233 0755 1E
1234 0756 0E
                                                                        push
                                                                         push
                                                                                     CS
1235 0757 1F
                                                                                     ds
 1236
1237 0758 B4 09
                                                                                                            ; DOS print string
                                                                         mov
                                                                                     ah, 9
1238 075A 52
1239 075B BA 09D9 R
1240 075E CD 21
                                                                         push
                                                                                     dx
                                                                                     dx, offset STRING_CRLF_DOS 21h
                                                                         mov
                                                                        pop
int
 1241 0760 5A
                                                                                     dx
 1242 0761 CD 21
                                                                                     21h
 1243 0763 1F
                                                                         POP
ASSUME
                                                                                    ds:FORMATTER_HEAP
1244
1245 0764 C3
1246 0765
                                                ret
PRINT_CRLF_STRING
1247
1248
1249 0765
                                                 DO_BEEP
                                                                                     dx, offset STRING_BEEP
PRINT_STRING_NO_CRLF
 1250 0765 BA 094F R
                                                                         mov
 1251 0768 E8 074C R
                                                                         call
1252 076B C3
1253 076C
                                                 DO_BEEP
                                                                                     ENDP
 1254
 1255
1256 076C
                                                 LOAD_CURSOR_POS
                                                                                     PROC NEAR
1257 076C 26: A0 00B0 R
1258 0770 26: 8A 26 00B1 R
1259 0775 C3
                                                                                     al, es:SAVED_COLUMN
ah, es:SAVED_ROW
                                                                        mov
                                                                         ret
 1260 0776
                                                 LOAD CURSOR POS
                                                                                     ENDP
1261
1262
 1263 0776
                                                 STORE_CURSOR_POS
                                                                                     PROC NEAR
1264 0776 26: A2 00B0 R
1265 077A 26: 88 26 00B1 R
1266 077F C3
                                                                        mov
                                                                                     es:SAVED_COLUMN, al
es:SAVED_ROW, ah
                                                                         ret
 1267 0780
                                                 STORE_CURSOR_POS
                                                                                     ENDP
1268
1269
 1270 0780
                                                 BAD_ENTRY_BEEP
                                                                                     PROC NEAR
1270 0780 E8 06D3 R
1271 0780 E8 06D3 R
1272 0783 E8 0765 R
1273 0786 26: FE 0E 00B1 R
1274 078B 26: C6 06 00B0 R 00
1275 0791 E8 06EE R
                                                                         call
                                                                                     READ_CURSOR_POS
                                                                                     DO_BEEP
                                                                         call
                                                                         dec
                                                                                     es:SAVED ROW
                                                                                     es:SAVED_COLUMN, 0
SET_CURSOR_POS
                                                                         call
 1276 0794 C3
                                                                         ret
1277 0795
1278
                                                 BAD_ENTRY_BEEP
                                                                                     FNDD
1279
 1280 0795
                                                 MOVE_CURSOR
                                                                                     PROC NEAR
1280 0795 E8 06D3 R
1281 0795 E8 06D3 R
1282 0798 26: A0 00B0 R
1283 079C 26: 8A 26 00B1 R
1284 07A1 2C 05
1285 07A3 26: A2 00B0 R
                                                                                     READ_CURSOR_POS
                                                                         call
                                                                                     al, es:SAVED COLUMN
                                                                         mov
                                                                         mov
sub
                                                                                     ah, es:SAVED_ROW
al, 5
                                                                                     es:SAVED COLUMN, al
                                                                         mov
1286 07A7 E8 06EE R
1287 07AA C3
1288 07AB
                                                                         call
                                                                                     SET_CURSOR_POS
                                                                         ret
                                                                                     ENDP
                                                 MOVE_CURSOR
1289
1290
1291 07AB
                                                                                     PROC NEAR
                                                 BIOS_WRITE_TEXT
 1292 07AB B4 0E
                                                                         mov
                                                                                     ah, OEh
10h
                                                                                                             : Write string
 1293 07AD CD 10
1294 07AF C3
                                                                         int
                                                                         ret
1295 07B0
                                                 BIOS_WRITE_TEXT
                                                                                     ENDP
1296
1297
1298 07B0
                                                 NUMBER_TO_HEX_DIGIT
                                                                                     PROC NEAR
 1299 07B0 3C 0A
1300 07B2 72 05
                                                                                     al, 10
short ZERO_TO_NINE
                                                                         cmp
jb
1301 07B4 2C 0A
                                                                         sub
                                                                                     al, 10
al, 'A'
 1302 07B6 04 41
                                                                         add
 1303 07B8 C3
1304 07B9
                                                 ZERO_TO_NINE:
 1305 07B9 04 30
                                                                                     al, '0'
                                                                         add
1306 07BB C3
1307 07BC
                                                 NUMBER_TO_HEX_DIGIT
                                                                                     ENDP
1308
1308
1309
1310 07BC 54 68 69 73 20 46
1311 4F 52 4D 41 54 20
1312 72 6F 75 74 69 65
1313 65 20 77 69 6C 6C
                                                 STRING_FORMAT_WILL_DESTROY
                                                                                                                         "This FORMAT routine will DESTROY"
1314 20 44 45 53 54 52
1315 4F 59
1316 07DC 20 41 4C 4C 20 54
1317 61 74 61 20 6F 6E
                                                                                                                         " ALL data on your disk!$"
               20 79 6F 75 72 20
64 69 73 6B 21 24
 1318
 1319
 1320 07F4 OA 45 6E 74 65 72
                                                 STRING_ENTER_DRIVE
                                                                                                                         "Enter drive # (0 or 1):$"
                                                                                                 DB
                                                                                                             OAh,
               20 64 72 69 76 65
20 23 20 28 30 20
6F 72 20 31 29 3A
 1321
 1322
1323
 1324
               24
 1325 080D 07 46 6F 72 6D 61
1326 74 20 66 61 69 6C
                                                 STRING_FORMAT_FAILED
                                                                                                                     7, "Format failed$"
 1326
1327
                65 64 24
1328 081C 0A 07 46 6C 61 67
1329 20 62 61 64 20 74
1330 72 61 63 6B 20 66
                                                 STRING_BAD_SECTOR_FAILED
                                                                                                 DB
                                                                                                             OAh, 7, "Flag bad track failed$"
1331 61 69 6C 65 64 24
1332 0834 0A 07 56 65 72 69
                                                 STRING_VERIFY_FAILED
                                                                                                 DB
                                                                                                             OAh, 7, "Verify failed$"
1332 0834 0A 07 56 65 72 69
1333 66 79 20 66 61 69
1334 6C 65 64 24
1335 0844 07 49 6E 69 74 20
1336 66 61 69 6C 75 72
1337 65 20 2D 20 61 62
1338 6F 72 74 65 84 24
1339 085C 0A 46 6F 72 6D 61
                                                 UNUSED_STRING_INIT_FAILURE
                                                                                                                    7, "Init failure - aborted$"
                                                STRING_FORMATTING
                                                                                                 DB
                                                                                                                        "Formatting...$"
                                                                                                             OAh.
1340 74 74 69 6E 67 2E
1341 2E 2E 24
1342 086B 46 6F 72 6D 61 74
                                                STRING_FORMAT_COMPLETE
                                                                                                DB
                                                                                                                         "Format complete$"
```

1344 65 74 65 24			
1345 087B 56 65 72 69 66 79 1346 69 6E 67 2E 2E 2E	STRING_VERIFYING	DB	"Verifying \$"
1347 20 20 24			
1348 088A 0A 56 65 72 69 66	STRING_VERIFY_COMPLETE	DB 0	DAh, "Verify complete\$"
1349 79 20 63 6F 6D 70			* * *
1350 6C 65 74 65 24			
1351 089B 4E 6F 20 6D 6F 72	STRING_NO_MORE_DEFECTS	DB	"No more defects accepted!\$"
1352 65 20 64 65 66 65 1353 63 74 73 20 61 63			
1353 63 74 73 20 61 63 1354 63 65 70 74 65 64			
1355 21 24			
1356 08B5 0A 41 6E 79 20 64	STRING_ANY_DEFECTS	DB 0	DAh, "Any defects (Y/N)? \$"
1357 65 66 65 63 74 73			• • • • • • • • • • • • • • • • • • • •
1358 20 28 59 2F 4E 29			
1359 3F 20 24			
1360 08CA 0A 28 50 72 65 73	STRING_PRESS_RET_TO_END	DB 0	OAh, "(Press <ret> to end defect list)\$"</ret>
1361 73 20 3C 52 45 54 1362 3E 20 74 6F 20 65			
1362 3E 20 74 6F 20 65 1363 6E 64 20 64 65 66			
1364 65 63 74 20 6C 69			
1365 73 74 29 24			
1366 08EC 43 59 4C 49 4E 44	STRING_CYLINDER	DB	"CYLINDER: \$"
1367 45 52 3A 20 20 20			
1368 20 20 24			
1369 08FB 20 20 20 20 48 45	STRING_HEAD	DB	" HEAD: \$"
1370 41 44 3A 20 20 20			
1371 20 20 24	CERTIFIC NOW MODE DESIGN SHEETING		Har (1/21/0 61
1372 090A 4D 6F 72 65 20 65 1373 6E 74 72 69 65 73	STRING_ASK_MORE_DEFECT_ENTRIES	DB	"More entries (Y/N)? \$"
1374 20 28 59 2F 4E 29			
1375 3F 20 24			
1376 091F 0A 41 72 65 20 79	STRING_CONFIRM_FORMAT	DB 0	DAh, "Are you SURE you want to format (Y/N)? \$"
1377 6F 75 20 53 55 52			
1378 45 20 79 6F 75 20			
1379 77 61 6E 74 20 74			
1380 6F 20 66 6F 72 6D			
1381 61 74 20 28 59 2F			
1382 4E 29 3F 20 24 1383 0948 20 20 20 20 20 20	EMPING SPACES	DB	" \$"
1384 24	STRING_SPACES	DB	. . .
1385 094F 07 24	STRING_BEEP	DB	7, "\$"
1386 0951 DB	01N1N0_D111		DDBh
1387 0952 6C			
		DB 6	5Ch
1388 0953 OA 50 72 65 73 73	STRING_PRESS_RET_TO_PROCEED		SCh DAh, "Press <ret> to proceed or <esc> to cancel\$"</esc></ret>
	STRING_PRESS_RET_TO_PROCEED		
1388 0953 0A 50 72 65 73 73 1389 20 3C 52 45 54 3E 1390 20 74 6F 20 70 72	STRING_PRESS_RET_TO_PROCEED		
1388 0953 0A 50 72 65 73 73 1389 20 3C 52 45 54 3E 1390 20 74 6F 20 70 72 1391 6F 63 65 65 54 20	STRING_PRESS_RET_TO_PROCEED		
1388 0953 0A 50 72 65 73 73 1389 20 3C 52 45 54 3E 1390 20 74 6F 20 70 72 1391 6F 63 65 65 54 20 1392 6F 72 20 3C 45 53	STRING_PRESS_RET_TO_PROCEED		
1388 0953 0A 50 72 65 73 73 13 1389 20 3C 52 45 54 3E 1390 20 74 6F 20 70 72 1391 6F 63 65 65 64 20 1392 6F 72 20 3C 45 53 1393 43 3E 20 74 6F 20	STRING_PRESS_RET_TO_PROCEED		
1388 0953 0A 50 72 65 73 73 1389 20 3C 52 45 54 5E 1390 20 74 6F 20 70 72 1391 6F 63 65 65 64 20 1392 6F 72 20 3C 45 53 1393 43 3E 20 74 6F 20 1394 63 61 6E 63 65 6C	STRING_PRESS_RET_TO_PROCEED		
1388 0953 0A 50 72 65 73 73 1389 20 3C 52 45 54 3E 1390 20 74 6F 20 70 72 1391 6F 63 65 65 6M 20 1392 6F 72 20 3C 45 53 1393 43 3E 20 74 6F 20 1394 63 61 6E 63 65 6C 1395 2E 2E 2E 24		O 8G	DAh, "Press <ret> to proceed or <esc> to cancel\$"</esc></ret>
1388 0953 0A 50 72 65 73 73 1389 20 3C 52 45 54 3E 1390 20 74 6F 20 70 72 1391 6F 63 65 65 54 20 1393 43 3E 20 74 6F 20 1394 63 61 6E 63 65 6C 1395 0981 50 72 65 73 73 20	STRING_PRESS_RET_TO_PROCEED STRING_ANY_KEY_TO_REBOOT		
1388 0953 0A 50 72 65 73 73 1389 20 3C 52 45 54 3E 1390 20 74 6F 20 70 72 1391 6F 63 65 65 6M 20 1392 6F 72 20 3C 45 53 1393 43 3E 20 74 6F 20 1394 63 61 6E 63 65 6C 1395 2E 2E 2E 24		O 8G	DAh, "Press <ret> to proceed or <esc> to cancel\$"</esc></ret>
1388 0953 0A 50 72 65 73 73 1389 20 3C 52 45 54 5E 1390 20 74 6F 20 70 72 1391 6F 63 65 65 64 20 1392 6F 72 20 3C 45 53 1393 43 3E 20 74 6F 20 1394 63 61 6E 63 65 6C 1395 2E 2E 2E 24 1396 0981 50 72 65 73 73 20 1397 61 6E 79 20 68 65		O 8G	DAh, "Press <ret> to proceed or <esc> to cancel\$"</esc></ret>
1388 0953 0A 50 72 65 73 73 1389 20 3C 52 45 54 5E 1390 20 74 6F 20 70 72 1391 6F 63 65 65 64 20 1392 6F 72 20 3C 45 53 1393 63 61 6E 63 65 6C 1395 2E 2E 2E 24 1396 0981 50 72 65 73 73 20 1397 61 6E 79 20 6E 65 1398 79 20 74 6F 20 72 1399 65 62 6F 6F 74 2E		O 8G	DAh, "Press <ret> to proceed or <esc> to cancel\$"</esc></ret>
1388 0953 0A 50 72 65 73 73 1389 20 3C 52 45 54 3E 1390 20 74 6F 20 70 72 1391 6F 63 65 65 68 20 1392 6F 72 20 3C 45 53 1393 43 3E 20 74 6F 20 1394 63 61 6E 63 65 6C 1395 0981 50 72 65 73 73 20 1397 61 6E 79 20 6B 65 1398 79 20 74 6F 20 72 1399 65 62 6F 6F 74 2E		O 8G	DAh, "Press <ret> to proceed or <esc> to cancel\$" "Press any key to reboot\$"</esc></ret>
1388 0953 0A 50 72 65 73 73 1389 20 3C 52 45 54 3E 1390 20 74 6F 20 70 72 1391 6F 63 65 65 68 20 1392 6F 72 20 3C 45 53 1393 43 3E 20 74 6F 20 1394 63 61 6E 63 65 6C 1395 2E 2E 2E 24 1396 0981 50 72 65 73 73 20 1397 61 6E 79 20 6E 65 1398 79 20 74 6F 20 72 1399 65 62 6F 6F 74 2E 1400 2E 2E 24 1401 099C 25 24	STRING_ANY_KEY_TO_REBOOT STRING_PERCENT STRING_SPACE	DB OB DB	UAh, "Press <ret> to proceed or <esc> to cancel\$" "Press any key to reboot\$" "%\$" " \$"</esc></ret>
1388 0953 0A 50 72 65 73 73 1389 20 3C 52 45 54 5E 1390 20 74 6F 20 70 72 1391 6F 63 65 65 64 20 1392 6F 72 20 3C 45 53 1393 43 3E 20 74 6F 20 1394 63 61 6E 63 65 6C 1395 2E 2E 2E 24 1396 0981 50 72 65 73 73 20 1397 61 6E 79 20 6E 65 1398 79 20 74 6F 20 72 1399 65 62 6F 6F 74 2E 1400 2E 2E 24 1401 099C 25 24 1402 099E 20 24 1403 099A0 07 57 72 69 74 65	STRING_ANY_KEY_TO_REBOOT STRING_PERCENT	DB 0	DAh, "Press <ret> to proceed or <esc> to cancel\$" "Press any key to reboot\$"</esc></ret>
1388 0953 0A 50 72 65 73 73 1389 20 3C 52 45 54 3E 1390 20 74 6F 20 70 72 1391 6F 63 65 65 58 20 1392 6F 72 20 3C 45 53 1393 43 3E 20 74 6F 20 1394 63 61 6E 63 65 6C 1395 0981 50 72 65 73 73 20 1397 61 6E 79 20 6B 65 1398 79 20 74 6F 20 72 1399 65 62 6F 6F 74 2E 1400 099C 25 24 1401 099C 25 24 1403 09A0 07 57 72 69 74 65 5	STRING_ANY_KEY_TO_REBOOT STRING_PERCENT STRING_SPACE	DB OB DB	UAh, "Press <ret> to proceed or <esc> to cancel\$" "Press any key to reboot\$" "%\$" " \$"</esc></ret>
1388 0953 0A 50 72 65 73 73 1389 20 3C 52 45 54 3E 1390 20 74 6F 20 70 72 1391 6F 63 65 65 68 20 1392 6F 72 20 3C 45 53 1393 43 3E 20 74 6F 20 1394 63 61 6E 63 65 6C 1395 2E 2E 2E 24 1396 0981 50 72 65 73 73 20 1397 61 6E 79 20 6B 65 1398 79 20 74 6F 20 72 1399 65 62 6F 6F 74 2E 1400 2E 2E 24 1401 099C 25 24 1403 099D 20 24 1403 099D 07 57 72 69 74 65 1404 20 62 75 66 66 65 1405 6C 6F 6F 6F 6F 74 6F	STRING_ANY_KEY_TO_REBOOT STRING_PERCENT STRING_SPACE	DB OB DB	UAh, "Press <ret> to proceed or <esc> to cancel\$" "Press any key to reboot\$" "%\$" " \$"</esc></ret>
1388 0953 0A 50 72 65 73 73 1389 20 3C 52 45 54 3E 1390 20 74 6F 20 70 72 1391 6F 63 65 65 84 20 1392 6F 72 20 3C 45 53 1393 43 3E 20 74 6F 20 1394 63 61 6E 63 65 6C 1395 2E 2E 2E 24 1396 0981 50 72 65 73 73 20 1397 61 6E 79 20 6B 65 1398 79 20 74 6F 20 72 1399 65 62 6F 6F 74 2E 1400 2E 2E 24 1402 099E 20 24 1402 099E 20 24 1403 09A0 07 57 72 69 74 65 1404 20 6E 75 66 66 65 1405 72 06 66 16 65	STRING_ANY_KEY_TO_REBOOT STRING_PERCENT STRING_SPACE STRING_BEEP_SECTOR_WRITE_FAIL	DB DB DB DB DB	### Press <ret> to proceed or <esc> to cancel\$" "Press any key to reboot\$" "%\$" " \$" 7, "Write buffer failed\$"</esc></ret>
1388 0953 0A 50 72 65 73 73 1389 20 3C 52 45 54 3E 1390 20 74 6F 20 70 72 1391 6F 63 65 65 68 20 1392 6F 72 20 3C 45 53 1393 43 3E 20 74 6F 20 1394 63 61 6E 63 65 6C 1395 2E 2E 2E 24 1396 0981 50 72 65 73 73 20 1397 61 6E 79 20 6B 65 1398 79 20 74 6F 20 72 1399 65 62 6F 6F 74 2E 1400 2E 2E 24 1401 099C 25 24 1403 099D 20 24 1403 099D 07 57 72 69 74 65 1404 20 62 75 66 66 65 1405 6C 6F 6F 6F 6F 74 6F	STRING_ANY_KEY_TO_REBOOT STRING_PERCENT STRING_SPACE	DB DB DB DB DB	### Press <ret> to proceed or <esc> to cancel\$" "Press any key to reboot\$" "%\$" " \$" 7, "Write buffer failed\$"</esc></ret>
1388 0953 0A 50 72 65 73 73 1389 20 3C 52 45 54 3E 1390 20 74 6F 20 70 72 1391 6F 63 65 65 58 20 1392 6F 72 20 3C 45 53 1393 43 3E 20 74 6F 20 1394 63 61 6E 63 65 6C 1395 0981 50 72 65 73 73 20 1397 61 6E 79 20 6B 65 1398 79 20 74 6F 20 72 1399 65 62 6F 6F 74 2E 1400 099E 20 24 1401 099C 25 24 1402 099E 20 24 1403 09A0 07 57 72 69 74 65 1404 20 62 75 66 66 65 1405 72 20 66 61 69 6C 1406 09E5 0A 49 6E 74 65 72	STRING_ANY_KEY_TO_REBOOT STRING_PERCENT STRING_SPACE STRING_BEEP_SECTOR_WRITE_FAIL	DB DB DB DB DB	### Press <ret> to proceed or <esc> to cancel\$" "Press any key to reboot\$" "%\$" " \$" 7, "Write buffer failed\$"</esc></ret>
1388 0953 0A 50 72 65 73 73 1389 20 3C 52 45 54 3E 1390 20 74 6F 20 70 72 1391 6F 63 65 65 5M 20 1392 6F 72 20 3C 45 53 1393 63 61 6E 63 65 6C 1395 0981 50 72 65 73 73 20 1397 61 6E 79 20 6B 65 1398 79 20 74 6F 20 72 1399 65 62 6F 6F 74 2E 1400 2E 2E 24 1402 099E 20 24 1402 099E 20 24 1403 09A0 07 57 72 69 74 65 1404 20 62 75 66 66 65 1405 72 20 66 61 69 6C 1406 65 6M 24 1407 09B5 0A 49 6E 74 65 72 1408 6C 65 61 76 65 20 1409 28 31 2D 31 35 29	STRING_ANY_KEY_TO_REBOOT STRING_PERCENT STRING_SPACE STRING_BEEP_SECTOR_WRITE_FAIL STRING_INTERLEAVE	DB OB DB DB DB DB	OAh, "Press <ret> to proceed or <esc> to cancel\$" "Press any key to reboot\$" "%\$" " \$" 7, "Write buffer failed\$" OAh, "Interleave (1-15):\$"</esc></ret>
1388 0953 0A 50 72 65 73 73 1389 20 3C 52 45 54 3E 1390 20 74 6F 20 70 72 1391 6F 63 65 65 84 20 1392 6F 72 20 3C 45 53 1393 43 3E 20 74 6F 20 1394 63 61 6E 63 65 6C 1395 2E 2E 2E 24 1396 0981 50 72 65 73 73 20 1397 61 6E 79 20 6E 65 1398 79 20 74 6F 20 72 1399 65 62 6F 6F 74 2E 1400 2E 2E 24 1401 099C 25 24 1403 099C 25 24 1403 099D 07 57 72 69 74 65 1404 20 62 75 66 66 65 1406 65 84 24 1407 09E5 0A 49 6E 74 65 72 1408 6C 65 61 69 6C 1406 62 65 61 69 6C 1406 62 65 61 65 62 1409 28 31 2D 31 35 29 1410 09C9 43 4F 4E 54 52 4F	STRING_ANY_KEY_TO_REBOOT STRING_PERCENT STRING_SPACE STRING_BEEP_SECTOR_WRITE_FAIL	DB DB DB DB DB	### Press <ret> to proceed or <esc> to cancel\$" "Press any key to reboot\$" "%\$" " \$" 7, "Write buffer failed\$"</esc></ret>
1388 0953 0A 50 72 65 73 73 1389 20 3C 52 45 54 3E 1390 20 74 6F 20 70 72 1391 6F 63 65 65 58 20 1392 6F 72 20 3C 45 53 1393 43 3E 20 74 6F 20 1394 63 61 6E 63 65 6C 1395 2E 2E 2E 24 1396 0981 50 72 65 73 73 20 1397 61 6E 79 20 76 6E 65 1398 79 20 74 6F 20 72 1399 65 62 6F 6F 74 2E 1400 099C 25 24 1400 099C 25 24 1402 099E 20 24 1404 209E 20 66 65 1405 72 20 66 61 69 6C 1405 72 20 66 61 69 6C 1406 65 61 76 65 20 1407 0985 0A 49 6E 74 65 72 1408 6C 65 61 76 65 20 1409 28 31 2D 31 35 29 1410 3A 24 1411 09C9 43 4F 4E 54 52 4F	STRING_ANY_KEY_TO_REBOOT STRING_PERCENT STRING_SPACE STRING_BEEP_SECTOR_WRITE_FAIL STRING_INTERLEAVE	DB OB DB DB DB DB	OAh, "Press <ret> to proceed or <esc> to cancel\$" "Press any key to reboot\$" "%\$" " \$" 7, "Write buffer failed\$" OAh, "Interleave (1-15):\$"</esc></ret>
1388 0953 0A 50 72 65 73 73 1389 20 3C 52 45 54 3E 1390 20 74 6F 20 70 72 1391 6F 63 65 65 58 20 1392 6F 72 20 3C 45 53 1393 63 61 6E 63 65 6C 1395 0981 50 72 65 73 73 20 1397 61 6E 79 20 6E 65 1398 79 20 74 6F 20 72 1399 65 62 6F 6F 74 2E 1400 2E 2E 24 1402 099E 20 24 1403 09A0 07 57 72 69 74 65 1404 20 62 75 66 66 65 1405 72 20 66 61 69 6C 1406 65 68 42 1407 09B5 0A 49 6E 74 65 72 1408 6C 65 61 76 65 20 1409 83 12 23 13 55 29 1410 3A 24 1411 09C9 43 4F 4E 54 52 4F 1411 09C9 43 4F 4E 54 52 4F	STRING_ANY_KEY_TO_REBOOT STRING_PERCENT STRING_SPACE STRING_BEEP_SECTOR_WRITE_FAIL STRING_INTERLEAVE STRING_CONTROL_BYTE_2	DB DB DB DB DB	DAh, "Press <ret> to proceed or <esc> to cancel\$" "Press any key to reboot\$" "%\$" " \$" 7, "Write buffer failed\$" DAh, "Interleave (1-15):\$" "CONTROL BYTE: 2\$"</esc></ret>
1388 0953 0A 50 72 65 73 73 1389 20 3C 52 45 54 3E 1390 20 74 6F 20 70 72 1391 6F 63 65 65 58 20 1392 6F 72 20 3C 45 53 1393 43 3E 20 74 6F 20 1394 63 61 6E 63 65 6C 1395 2E 2E 2E 24 1396 0981 50 72 65 73 73 20 1397 61 6E 79 20 6B 65 1398 79 20 74 6F 20 72 1399 65 62 6F 6F 74 2E 1400 2E 2E 24 1402 099E 20 24 1402 099E 20 24 1403 09A0 07 57 72 69 74 65 1404 20 6E 75 66 66 65 1405 72 20 66 61 69 6C 1405 65 87 24 1407 09B5 0A 49 6E 74 65 72 1408 6C 65 61 76 65 20 1409 28 31 2D 31 35 29 1410 3A 24 1411 09C9 43 4F 4E 54 52 4F 1411 09C9 43 4F 4E 54 52 4F 1412 4C 20 42 59 54 45 1414 09D9 0D 0A 24	STRING_ANY_KEY_TO_REBOOT STRING_PERCENT STRING_SPACE STRING_BEEP_SECTOR_WRITE_FAIL STRING_INTERLEAVE	DB OB DB DB DB DB	DAh, "Fress <ret> to proceed or <esc> to cancel\$" "Press any key to reboot\$" "\$\$" "\$" 7, "Write buffer failed\$" DAh, "Interleave (1-15):\$" "CONTROL BYTE: 2\$"</esc></ret>
1388 0953 0A 50 72 65 73 73 1389 20 3C 52 45 54 3E 1390 20 74 6F 20 70 72 1391 6F 63 65 65 58 20 1392 6F 72 20 3C 45 53 1393 43 3E 20 74 6F 20 1394 63 61 6E 63 65 6C 1395 0881 50 72 65 73 73 20 1397 61 6E 79 20 6B 65 1398 79 20 74 6F 20 72 1399 65 62 6F 6F 74 2E 1400 099C 25 24 1402 099E 20 24 1403 09A0 07 57 72 69 74 65 1405 72 20 66 61 69 6C 1405 72 20 66 61 69 6C 1406 65 67 24 1407 09B5 0A 49 6E 74 65 72 1408 6C 65 61 76 65 20 1409 28 31 2D 31 35 29 1410 3A 24 1411 09C9 45 74 62 4F 1412 4C 20 42 59 54 45 1413 3A 20 32 24 1414 09D9 0D 0A 24 1415 09DC 00 08 10 00 08	STRING_ANY_KEY_TO_REBOOT STRING_PERCENT STRING_SPACE STRING_BEEP_SECTOR_WRITE_FAIL STRING_INTERLEAVE STRING_CONTROL_BYTE_2	DB OB DB DB DB DB	DAh, "Press <ret> to proceed or <esc> to cancel\$" "Press any key to reboot\$" "%\$" " \$" 7, "Write buffer failed\$" DAh, "Interleave (1-15):\$" "CONTROL BYTE: 2\$"</esc></ret>
1388 0953 0A 50 72 65 73 73 1389 20 3C 52 45 54 3E 1390 20 74 6F 20 70 72 1391 6F 63 65 65 58 20 1392 6F 72 20 3C 45 53 1393 43 3E 20 74 6F 20 1394 63 61 6E 63 65 6C 1395 2E 2E 2E 24 1396 0981 50 72 65 73 73 20 1397 61 6E 79 20 6B 65 1398 79 20 74 6F 20 72 1399 65 62 6F 6F 74 2E 1400 2E 2E 24 1402 099E 20 24 1402 099E 20 24 1403 09A0 07 57 72 69 74 65 1404 20 6E 75 66 66 65 1405 72 20 66 61 69 6C 1405 65 87 24 1407 09B5 0A 49 6E 74 65 72 1408 6C 65 61 76 65 20 1409 28 31 2D 31 35 29 1410 3A 24 1411 09C9 43 4F 4E 54 52 4F 1411 09C9 43 4F 4E 54 52 4F 1412 4C 20 42 59 54 45 1414 09D9 0D 0A 24	STRING_ANY_KEY_TO_REBOOT STRING_PERCENT STRING_SPACE STRING_BEEP_SECTOR_WRITE_FAIL STRING_INTERLEAVE STRING_CONTROL_BYTE_2	DB OB DB DB DB DB	DAh, "Fress <ret> to proceed or <esc> to cancel\$" "Press any key to reboot\$" "\$\$" "\$" 7, "Write buffer failed\$" DAh, "Interleave (1-15):\$" "CONTROL BYTE: 2\$"</esc></ret>
1388 0953 0A 50 72 65 73 73 1389 20 3C 52 45 54 3E 1390 20 74 6F 20 70 72 1391 6F 63 65 65 84 20 1392 6F 72 20 3C 45 53 1393 43 3E 20 74 6F 20 1394 63 61 6E 63 65 6C 1395 0981 50 72 65 73 73 20 1397 61 6E 79 20 6E 65 1398 79 20 74 6F 20 72 1399 65 62 6F 6F 74 2E 1400 2E 2E 24 1401 099E 25 24 1402 099E 20 24 1403 09A0 07 57 72 69 74 65 1404 20 62 75 66 66 65 1404 20 62 75 66 66 65 1405 72 20 66 61 69 6C 1406 68 78 24 1410 09E5 78 78 78 78 78 78 1407 09E5 78 78 78 78 78 1408 6C 65 61 76 65 20 1409 28 31 20 31 35 29 1410 3A 24 1411 09C9 43 4F 4E 54 52 4F 1411 09C9 43 4F 4E 54 52 4F 1413 09D0 00 08 10 00 08 00 1416 098 07 02 04 04 04	STRING_ANY_KEY_TO_REBOOT STRING_PERCENT STRING_SPACE STRING_BEEP_SECTOR_WRITE_FAIL STRING_INTERLEAVE STRING_CONTROL_BYTE_2	DB OB DB DB DB DB	DAh, "Fress <ret> to proceed or <esc> to cancel\$" "Press any key to reboot\$" "\$\$" "\$" 7, "Write buffer failed\$" DAh, "Interleave (1-15):\$" "CONTROL BYTE: 2\$"</esc></ret>
1388 0953 0A 50 72 65 73 73 1389 20 3C 52 45 54 3E 1390 20 74 6F 20 70 72 1391 6F 63 65 65 58 20 1392 6F 72 20 3C 45 53 1393 43 3E 20 74 6F 20 1394 63 61 6E 63 65 6C 1395 2E 2E 2E 24 1396 0981 50 72 65 73 73 20 1397 61 6E 79 20 6E 65 1398 79 20 74 6F 20 72 1399 65 62 6F 6F 74 2E 1400 099C 25 24 1402 099E 20 24 1402 099E 20 24 1404 20 6E 75 66 66 65 1405 72 20 66 61 69 6C 1405 72 20 66 61 69 6C 1406 65 61 76 65 20 1407 0985 0A 49 6E 74 65 72 1408 66 66 65 61 76 65 20 1409 28 31 2D 31 35 29 1410 3A 24 1411 09C9 43 4F 4E 54 52 4F 1411 09C9 43 4F 4E 54 52 4F 1412 4C 20 42 59 54 45 1413 09D0 00 81 00 08 00 1416 08 07 02 04 04 04 1417 09D0 00 00 00	STRING_ANY_KEY_TO_REBOOT STRING_PERCENT STRING_SPACE STRING_BEEP_SECTOR_WRITE_FAIL STRING_INTERLEAVE STRING_CONTROL_BYTE_2 STRING_CRLF_DOS	DB D	DAh, "Fress <ret> to proceed or <esc> to cancel\$" "Press any key to reboot\$" "\$\$" "\$" 7, "Write buffer failed\$" DAh, "Interleave (1-15):\$" "CONTROL BYTE: 2\$" DDh, OAh, "\$" 0, 8, 10h, 0, 8, 0, 8, 7, 2, 4, 4, 4, 0, 0, 0, 0</esc></ret>
1388 0953 0A 50 72 65 73 73 1389 20 3C 52 45 54 3E 1390 20 74 6F 20 70 72 1391 6F 63 65 65 58 20 1392 6F 72 20 3C 45 53 1393 43 3E 20 74 6F 20 1394 63 61 6E 63 65 6C 1395 0981 50 72 65 73 73 20 1397 61 6E 79 20 6B 65 1398 79 20 74 6F 20 72 1399 65 62 6F 6F 74 2E 1400 099E 20 24 1400 099E 20 26 1400 6C 65 61 66 65 1405 72 20 66 61 69 6C 1406 65 67 44 1407 0985 0A 49 6E 74 65 72 1408 6C 65 61 76 65 20 1409 28 31 20 31 35 29 1410 3A 24 1411 09C9 0A 24 1411 09C9 0A 24 1411 09D9 0A 24 1411 09D9 0A 24 1411 09D0 0A 24 1411 09D0 0D 0A 24 1415 09DC 0D 0B 10 0D 0D 1416 08 07 02 04 04 04 1417 00 00 00 00 00 1416 08 07 02 04 04 04 1417 000 00 00 00 00	STRING_ANY_KEY_TO_REBOOT STRING_PERCENT STRING_SPACE STRING_BEEP_SECTOR_WRITE_FAIL STRING_INTERLEAVE STRING_CONTROL_BYTE_2 STRING_CRLF_DOS	DB D	DAh, "Fress <ret> to proceed or <esc> to cancel\$" "Press any key to reboot\$" "\$\$" "\$" 7, "Write buffer failed\$" DAh, "Interleave (1-15):\$" "CONTROL BYTE: 2\$" DDh, OAh, "\$" 0, 8, 10h, 0, 8, 0, 8, 7, 2, 4, 4, 4, 0, 0, 0, 0</esc></ret>
1388 0953 0A 50 72 65 73 73 1389 20 3C 52 45 54 3E 1390 20 74 6F 20 70 72 1391 6F 63 65 65 58 20 1392 6F 72 20 3C 45 53 1393 63 61 6E 63 65 6C 1393 61 6E 79 20 6E 65 1395 62 62 62 62 62 62 62 62 62 62 62 62 62	STRING_ANY_KEY_TO_REBOOT STRING_PERCENT STRING_SPACE STRING_BEEP_SECTOR_WRITE_FAIL STRING_INTERLEAVE STRING_CONTROL_BYTE_2 STRING_CRLF_DOS	DB D	DAh, "Fress <ret> to proceed or <esc> to cancel\$" "Press any key to reboot\$" "\$\$" "\$" 7, "Write buffer failed\$" DAh, "Interleave (1-15):\$" "CONTROL BYTE: 2\$" DDh, OAh, "\$" 0, 8, 10h, 0, 8, 0, 8, 7, 2, 4, 4, 4, 0, 0, 0, 0</esc></ret>
1388 0953 0A 50 72 65 73 73 1389 20 3C 52 45 54 3E 1390 20 74 6F 20 70 72 1391 6F 63 65 65 84 20 1392 6F 72 20 3C 45 53 1393 43 3E 20 74 6F 20 1394 63 61 6E 63 65 6C 1395 2E 2E 2E 24 1396 0981 50 72 65 73 73 20 1397 61 6E 79 20 6B 65 1398 79 20 74 6F 20 72 1399 65 62 6F 6F 74 2E 1400 2E 2E 24 1402 099E 20 24 1403 090C 07 57 72 69 74 65 1404 20 6E 75 66 66 65 1405 72 20 66 61 69 6C 1405 72 20 66 61 69 6C 1406 65 67 24 1407 0995 0A 49 6E 74 65 72 1408 6C 65 61 76 65 20 1409 28 31 2D 31 35 29 1410 3A 24 1411 099C 43 4F 4E 54 52 4F 1412 9C 4C 20 42 59 54 45 1413 099C 00 08 10 00 08 00 1418 099C 50 72 6F 63 65 73 1419 73 69 6E 67 20 40 44 1417 00 00 00 00 00 1418 099C 50 72 6F 63 65 73 1419 73 69 6E 67 20 64 1410 73 69 6E 67 20 64 1417 73 69 6E 67 20 65 41 1418 099C 70 6F 63 65 73 1419 73 69 6E 67 20 64 1410 73 69 6E 67 20 65 1419 73 69 6E 67 20 65 1419 73 69 6E 67 20 65 1420 65 66 65 63 74 73 1421 2E 2E 2E 2E 44	STRING_ANY_KEY_TO_REBOOT STRING_PERCENT STRING_SPACE STRING_BEEP_SECTOR_WRITE_FAIL STRING_INTERLEAVE STRING_CONTROL_BYTE_2 STRING_CRLF_DOS STRING_PROCESSING_DEFECTS	DB D	DAh, "Fress <ret> to proceed or <esc> to cancel\$" "Press any key to reboot\$" "\$;" 7, "Write buffer failed\$" DAh, "Interleave (1-15):\$" "CONTROL BYTE: 2\$" DDh, OAh, "\$" 0, 8, 10h, 0, 8, 0, 8, 7, 2, 4, 4, 4, 0, 0, 0, 0 </esc></ret>
1388 0953 0A 50 72 65 73 73 1389 20 3C 52 45 54 3E 1390 20 74 6F 20 70 72 1391 6F 63 65 65 58 20 1392 6F 72 20 3C 45 53 1393 43 3E 20 74 6F 20 1394 63 61 6E 63 65 6C 1395 0881 50 72 65 73 73 20 1397 61 6E 79 20 6E 65 1398 79 20 74 6F 20 72 1399 65 62 6F 6F 74 2E 1400 099C 25 24 1400 099C 25 24 1400 099C 20 24 1401 099C 25 24 1402 099E 20 24 1404 209E 20 66 65 1405 72 20 66 66 65 1405 72 20 66 66 65 1405 72 20 66 66 65 1406 65 67 64 24 1407 09B5 0A 49 6E 74 65 72 1408 6C 65 61 76 65 20 1409 28 31 2D 31 35 29 1410 3A 24 1411 09C9 43 4F 4E 54 52 4F 1412 4C 20 42 59 54 45 1413 3A 20 32 24 1414 09PO 0D 0A 24 1415 09DC 00 08 10 00 08 00 1416 08 07 02 04 04 04 1417 09 0D 00 02 1418 09EC 50 72 6F 63 65 73 1419 73 69 6E 67 00 E3 1420 1422 12E 2E 2E 24	STRING_ANY_KEY_TO_REBOOT STRING_PERCENT STRING_SPACE STRING_BEEP_SECTOR_WRITE_FAIL STRING_INTERLEAVE STRING_CONTROL_BYTE_2 STRING_CRLF_DOS	DB D	DAh, "Fress <ret> to proceed or <esc> to cancel\$" "Press any key to reboot\$" "\$\$" "\$" 7, "Write buffer failed\$" DAh, "Interleave (1-15):\$" "CONTROL BYTE: 2\$" DDh, OAh, "\$" 0, 8, 10h, 0, 8, 0, 8, 7, 2, 4, 4, 4, 0, 0, 0, 0</esc></ret>
1388 0953 0A 50 72 65 73 73 1389 20 3C 52 45 54 3E 1390 20 74 6F 20 70 72 1391 6F 63 65 65 84 20 1392 6F 72 20 3C 45 53 1393 63 61 6E 63 65 6C 1393 63 61 6E 63 65 6C 1393 67 62 6E 77 2 20 3C 45 73 1397 61 6E 79 20 6E 65 1398 79 20 74 6F 20 72 1399 65 62 6F 6F 74 2E 1399 65 62 6F 6F 74 2E 1400 099E 20 24 1400 309A 07 57 72 69 74 65 1405 72 20 66 61 69 6C 1406 65 68 1 26 65 65 1405 72 20 66 61 69 6C 1406 65 68 1 26 65 1407 0985 0A 49 6E 74 65 72 1408 6C 65 61 76 65 20 1409 28 31 2D 31 35 29 1410 3A 24 1411 09C9 43 4F 4E 52 4F 1412 4C 20 42 59 54 45 1413 3A 20 32 24 1414 09D9 0D 0A 24 1411 09C9 43 4F 4E 52 4F 1412 4C 20 42 59 54 45 1413 3A 20 32 24 1414 09D9 0D 0A 24 1411 09C0 00 8 10 00 08 00 1416 08 07 02 04 04 04 1417 00 00 00 00 1418 09CC 50 72 6F 63 65 73 1419 73 69 6E 67 20 58 1422 2E 2E 22 4 1423 0A02 28 63 29 43 6F 70 1424 79 72 69 67 68 74	STRING_ANY_KEY_TO_REBOOT STRING_PERCENT STRING_SPACE STRING_BEEP_SECTOR_WRITE_FAIL STRING_INTERLEAVE STRING_CONTROL_BYTE_2 STRING_CRLF_DOS STRING_PROCESSING_DEFECTS	DB D	DAh, "Fress <ret> to proceed or <esc> to cancel\$" "Press any key to reboot\$" "\$;" 7, "Write buffer failed\$" DAh, "Interleave (1-15):\$" "CONTROL BYTE: 2\$" DDh, OAh, "\$" 0, 8, 10h, 0, 8, 0, 8, 7, 2, 4, 4, 4, 0, 0, 0, 0 </esc></ret>
1388 0953 0A 50 72 65 73 73 1389 20 3C 52 45 54 3E 1390 20 74 6F 20 70 72 1391 6F 63 65 65 65 42 20 1392 6F 72 20 3C 45 53 1393 43 3E 20 74 6F 20 1394 63 61 6E 63 65 6C 1395 2E 2E 2E 24 1396 0981 50 72 65 73 73 20 1397 61 6E 79 20 6E 65 1398 79 20 74 6F 20 72 1399 65 62 6F 6F 74 2E 1400 2E 2E 24 1402 099E 20 24 1402 099E 20 24 1404 20 62 75 66 66 65 1405 72 20 66 61 65 65 1405 72 20 66 61 65 65 1406 65 68 24 1407 0995 0A 49 6E 74 65 72 1408 66 65 61 76 65 20 1409 28 31 2D 31 35 29 1410 3A 24 1411 09C9 43 4F 4E 54 52 4F 1411 09C9 43 4F 4E 54 52 4F 1412 4C 20 42 59 54 45 1413 09C9 00 00 00 1416 08 07 02 04 04 04 1417 09D0 00 00 00 1418 09EC 50 72 6F 63 65 73 1419 73 69 6E 67 20 64 1410 1416 09 00 00 00 1416 08 07 02 04 04 04 1417 00 00 00 00 00 1418 09EC 50 72 6F 63 65 73 1419 73 69 6E 67 20 65 1422 1422 1423 0A02 28 63 29 43 6F 70 1424 79 72 69 67 74 1425 77 26 9 76 87 74	STRING_ANY_KEY_TO_REBOOT STRING_PERCENT STRING_SPACE STRING_BEEP_SECTOR_WRITE_FAIL STRING_INTERLEAVE STRING_CONTROL_BYTE_2 STRING_CRLF_DOS STRING_PROCESSING_DEFECTS	DB D	DAh, "Fress <ret> to proceed or <esc> to cancel\$" "Press any key to reboot\$" "\$;" 7, "Write buffer failed\$" DAh, "Interleave (1-15):\$" "CONTROL BYTE: 2\$" DDh, OAh, "\$" 0, 8, 10h, 0, 8, 0, 8, 7, 2, 4, 4, 4, 0, 0, 0, 0 </esc></ret>
1388 0953 0A 50 72 65 73 73 1389 20 3C 52 45 54 3E 1390 20 74 6F 20 70 72 1391 6F 63 65 65 58 20 1392 6F 72 20 3C 45 53 1393 43 3E 20 74 6F 20 1394 63 61 6E 63 65 6C 1395 0881 50 72 65 73 73 20 1397 61 6E 79 20 6E 65 1398 79 20 74 6F 20 72 1399 65 62 6F 6F 74 2E 1400 099C 25 24 1400 099C 25 24 1400 099C 20 24 1400 099C 27 66 66 66 65 1405 72 20 66 61 69 6C 1406 65 67 20 66 66 65 1405 72 20 66 61 69 6C 1407 0985 0A 49 6E 74 65 72 1408 6C 65 61 76 65 20 1409 28 31 20 31 35 29 1410 3A 24 1411 09C9 43 4F 4E 54 52 4F 1411 09C9 43 4F 4E 54 52 4F 1412 4C 20 42 59 54 45 1413 3A 20 32 24 1414 09D9 0D 0A 24 1416 08 07 02 04 04 04 1416 08 07 02 04 04 04 1417 00 00 00 00 00 1416 08 07 02 04 04 04 1417 00 00 00 00 00 1418 09EC 50 72 6F 63 65 73 1419 73 69 6E 67 20 6A 1419 73 69 6E 67 20 6A 1410 1420 65 66 65 63 73 1419 73 69 6E 67 20 6A 1410 73 69 6E 67 20 6A 1410 73 69 6E 67 70 6A 1421 2E 2E 2E 24 1422 0A02 28 63 29 43 6F 70 1424 79 72 69 67 68 74 1425 20 31 33 33 37 20	STRING_ANY_KEY_TO_REBOOT STRING_PERCENT STRING_SPACE STRING_BEEP_SECTOR_WRITE_FAIL STRING_INTERLEAVE STRING_CONTROL_BYTE_2 STRING_CRLF_DOS STRING_PROCESSING_DEFECTS	DB D	DAh, "Press <ret> to proceed or <esc> to cancel\$" "Press any key to reboot\$" "\$\$" "\$" 7, "Write buffer failed\$" DAh, "Interleave (1-15):\$" "CONTROL BYTE: 2\$" DDh, OAh, "\$" 0, 8, 10h, 0, 8, 0, 8, 7, 2, 4, 4, 4, 0, 0, 0, 0 "Processing defects\$" "(c) Copyright 1987 SMS"</esc></ret>
1388 0953 0A 50 72 65 73 73 1389 20 3C 52 45 54 3E 1390 20 74 6F 20 70 72 1391 6F 63 65 65 65 42 20 1392 6F 72 20 3C 45 53 1393 43 3E 20 74 6F 20 1394 63 61 6E 63 65 6C 1395 2E 2E 2E 24 1396 0981 50 72 65 73 73 20 1397 61 6E 79 20 6E 65 1398 79 20 74 6F 20 72 1399 65 62 6F 6F 74 2E 1400 2E 2E 24 1402 099E 20 24 1402 099E 20 24 1404 20 62 75 66 66 65 1405 72 20 66 61 65 65 1405 72 20 66 61 65 65 1406 65 68 24 1407 0995 0A 49 6E 74 65 72 1408 66 65 61 76 65 20 1409 28 31 2D 31 35 29 1410 3A 24 1411 09C9 43 4F 4E 54 52 4F 1411 09C9 43 4F 4E 54 52 4F 1412 4C 20 42 59 54 45 1413 09C9 00 00 00 1416 08 07 02 04 04 04 1417 09D0 00 00 00 1418 09EC 50 72 6F 63 65 73 1419 73 69 6E 67 20 64 1410 1416 09 00 00 00 1416 08 07 02 04 04 04 1417 00 00 00 00 00 1418 09EC 50 72 6F 63 65 73 1419 73 69 6E 67 20 65 1422 1422 1423 0A02 28 63 29 43 6F 70 1424 79 72 69 67 74 1425 77 26 9 76 87 74	STRING_ANY_KEY_TO_REBOOT STRING_PERCENT STRING_SPACE STRING_BEEP_SECTOR_WRITE_FAIL STRING_INTERLEAVE STRING_CONTROL_BYTE_2 STRING_CRLF_DOS STRING_PROCESSING_DEFECTS	DB O	### Press <pre> "Press any key to reboot\$" "Press any key to reboot\$" "\$\$" "\$" 7, "Write buffer failed\$" DAh, "Interleave (1-15):\$" "CONTROL BYTE: 2\$" DDh, OAh, "\$" 0, 8, 10h, 0, 8, 0, 8, 7, 2, 4, 4, 4, 0, 0, 0, 0 "Processing defects\$" "(c) Copyright 1987 SMS" "Yeah, do not disassemble or anything." </pre>
1388 0953 0A 50 72 65 73 73 1389 20 3C 52 45 54 3E 1390 20 74 6F 20 70 72 1391 6F 63 65 65 68 20 1392 6F 72 20 3C 45 53 1393 43 3E 20 74 6F 20 1394 63 61 6E 63 65 6C 1395 2E 2E 2E 24 1396 0981 50 72 65 73 73 20 1397 61 6E 79 20 6E 65 1398 79 20 74 6F 20 72 1399 65 62 6F 6F 74 2E 1400 2E 2E 24 1401 099C 25 24 1402 099E 20 24 1403 09A0 07 57 72 69 74 65 1404 20 6E 75 66 66 65 1405 72 20 66 61 66 65 1405 72 20 66 61 76 65 20 1409 28 31 2D 31 35 29 1410 09C 28 31 2D 31 35 29 1410 09C 28 31 2D 31 35 29 1410 09C 38 34 4E 54 52 4F 1411 09C9 43 4F 4E 54 52 4F 1412 4C 20 42 59 54 45 1414 09D9 0D 0A 24 1415 09DC 00 08 10 00 08 00 1416 08 07 02 04 04 04 1417 00 00 00 00 00 1418 09EC 50 72 6F 63 65 73 1419 73 69 6E 67 20 6B 1420 65 66 65 63 73 74 73 1421 2E 2E 2E 24 1422 1423 0A02 28 63 29 43 6F 70 1424 79 72 69 67 68 74 1425 20 31 39 38 37 20 1427	STRING_ANY_KEY_TO_REBOOT STRING_PERCENT STRING_SPACE STRING_BEEP_SECTOR_WRITE_FAIL STRING_INTERLEAVE STRING_CONTROL_BYTE_2 STRING_CRLF_DOS STRING_PROCESSING_DEFECTS	DB O	DAh, "Press <ret> to proceed or <esc> to cancel\$" "Press any key to reboot\$" "\$\$" "\$" 7, "Write buffer failed\$" DAh, "Interleave (1-15):\$" "CONTROL BYTE: 2\$" DDh, OAh, "\$" 0, 8, 10h, 0, 8, 0, 8, 7, 2, 4, 4, 4, 0, 0, 0, 0 "Processing defects\$" "(c) Copyright 1987 SMS"</esc></ret>
1388 0953 0A 50 72 65 73 73 1389 20 3C 52 45 54 3E 1390 20 74 6F 20 70 72 1391 6F 63 65 65 58 20 1392 6F 72 20 3C 45 53 1393 43 3E 20 74 6F 20 1394 63 61 6E 63 65 6C 1395 2E 2E 2E 24 1396 0981 50 72 65 73 73 20 1397 61 6E 79 20 6E 65 1398 79 20 74 6F 20 72 1399 65 62 6F 6F 74 2E 1400 099C 25 24 1402 099E 20 24 1404 099C 25 24 1404 20 6E 75 66 66 65 1405 72 20 66 61 69 6C 1406 65 61 66 65 20 1407 0985 0A 49 6E 74 65 72 1408 66 66 65 61 76 65 20 1409 28 31 2D 31 35 29 1410 3A 24 1411 09C9 43 4F 4E 54 52 4F 1411 09C9 43 4F 4E 54 52 4F 1412 4C 20 42 59 54 45 1413 0900 00 00 00 1416 08 07 02 04 04 04 1417 0900 00 00 00 1416 08 07 02 04 04 04 1417 0900 00 00 00 1416 08 07 02 04 04 04 1417 00 00 00 00 00 1418 09EC 50 72 6F 63 65 73 1419 73 69 6E 67 20 54 1422 1422 1423 0A02 28 63 29 43 6F 70 1424 79 72 69 67 68 74 1425 1421 39 38 37 20 1426 53 4D 53	STRING_ANY_KEY_TO_REBOOT STRING_PERCENT STRING_SPACE STRING_BEEP_SECTOR_WRITE_FAIL STRING_INTERLEAVE STRING_CONTROL_BYTE_2 STRING_CRLF_DOS STRING_PROCESSING_DEFECTS	DB O	### Press <pre> "Press any key to reboot\$" "Press any key to reboot\$" "\$\$" "\$" 7, "Write buffer failed\$" DAh, "Interleave (1-15):\$" "CONTROL BYTE: 2\$" DDh, OAh, "\$" 0, 8, 10h, 0, 8, 0, 8, 7, 2, 4, 4, 4, 0, 0, 0, 0 "Processing defects\$" "(c) Copyright 1987 SMS" "Yeah, do not disassemble or anything." </pre>

```
1431
                                                  PAGE
1432
1433
1434
                                                              Option ROM initialization
1435
1436 OA17
1437 OA17 E8 1141 R
                                                  ROM_INIT
                                                                                        PROC FAR
ZERO_DS_ALT
                                                                           ASSUME ds: ZEROSEG
1438
1438
1440 OA1A FA
1441 OA1B C7 O6 0064 R 1148 R
1442 OA21 8C OE 0066 R
1443 OA25 C7 O6 0104 R 0020 R
                                                                           cli
                                                                                        INT19H_OFFSET, offset INT19H_HANDLER
                                                                           mov
                                                                                        INT19H_SEGMENT, cs
INT41H_OFFSET, offset INT41H_DATA
INT41H_SEGMENT, cs
1444 0A2B 8C 0E 0106 R
                                                                           mov
1445 OA2F 06
1446 OA30 57
                                                                           push
push
1447 0A31 C4 3E 004C R
1448 0A35 89 3E 0100 R
1449 0A39 8C 06 0102 R
1450
                                                                           les
                                                                                        di, dword ptr INT13H_OFFSET
                                                                                       INT40H_OFFSET, di
INT40H_SEGMENT, es
es:ZEROSEG
di
                                                                           ASSUME
1451 0A3D 5F
1452 0A3E 07
1453 0A3F C7 06 004C R 0B77 R
1454 0A45 8C 0E 004E R
                                                                                        es
INT13H_OFFSET, offset INT13H_HANDLER
                                                                           pop
                                                                           mov
1455 OA49 FB
1456
                                                                            sti
1457 0A4A E8 1139 R
                                                                           call
                                                                                        ZERO DS
1458
                                                                           ASSUME
                                                                                        ds: ZEROSEG
1459 OA4D C6 O6 0475 R 00
                                                                                        BDA_NUMBER_OF_HARD_DISKS, 0
                                                                           mov
1460
1461 OA52 FA
1462 OA53 E4 21
1463 OA55 24 FE
                                                                           cli
                                                                                        al, 21h
                                                                           in
                                                                           and
                                                                                                                                         ; Unmask IRQ0 (timer)
                                                                                        al, OFEh
1464 OA57 E6 21
1465 OA59 FB
1466
                                                                           out
sti
                                                                                        21h, al
1467 0A5A E8 0D61 R
                                                                           call
                                                                                        MASTER RESET
                                                                                        master_reset
ah, inti3H_OP_14_CONTROLLER_DIAGS
DO_CALL_INTI3H
short CONTROLLER_DIAGS_FINE
1468 0A5D B4 14
1469 0A5F E8 0B5A R
                                                                           mov
call
                                                                           jnc
mov
1470 0A62 73 06
1470 0A62 73 06
1471 0A64 BE 1216 R
1472 0A67 E9 0B00 R
                                                                                        si, offset STRING_B
DISK_INIT_ERROR
                                                 jmp
CONTROLLER_DIAGS_FINE:
1473 OA6A
1474 0A6A B4 12
1475 0A6C E8 0B5A R
1476 0A6F 73 06
1477 0A71 BE 1214 R
1478 0A74 E9 0B00 R
                                                                                        ah, INT13H_OP_12_SRAM_DIAGS
DO_CALL_INT13H
short SRAM_DIAGS_FINE
                                                                           call
                                                                           jnc
mov
                                                                                        si, offset STRING_A
DISK_INIT_ERROR
                                                                           jmp
1479 0A77
                                                  SRAM DIAGS FINE:
1480 0A77 50
1481 0A78 53
1482 0A79 51
                                                                           push
                                                                           push
                                                                                        CX
1483 0A7A 52
1484 0A7B B8 01C2
1485 0A7E 81 3E 0472 R 04D2
                                                                           push
mov
                                                                                                                                         ; Drive ready (spin-up) timeout
; If the soft reset flag is not present
; we keep the long timeout of 450,
; otherwise we shorten it to 90
                                                                                        BDA SOFT RESET FLAG. 4D2h
                                                                           cmp
1486 0A84 75 03
1487 0A86 B8 005A
1488 0A89
                                                                                        short NOT_SOFT_RESET_4D2
ax, 90
                                                  NOT_SOFT_RESET_4D2:
1489 0A89 E8 1028 R
                                                                           call
                                                                                        BDA_TIMER_TO_CX_DX
1490 OA8C 03 D0
1491 OA8E 83 D1 00
                                                                                        dx, ax
cx, ⊪
                                                                           adc
1492 0A91 8B D9
                                                                           mov
                                                                                        bx. cx
1493 0A93 8B C2
1494
1495 0A95
                                                 DO_TEST_READY:
1496 0A95 50
1497 0A96 53
                                                                           push
                                                                                        ah, INT13H_OP_10_TEST_DRIVE_READY
1498 0A97 B4 10
                                                                           mov
1499 0A99 E8 0B5A R
1499 0A99 E8 0B5A R
1500 0A9C 73 IE
1501 0A9E 80 FC 27
1502 0AA1 74 I9
1503 0AA3 E8 1028 R
1504 0AA6 5B
                                                                                        DO_CALL_INT13H
short DO_RECALIBRATE
                                                                            call
                                                                            jnc
                                                                                        ah, INTI3H_STATUS_27_NEED_RECALIBRATE
short D0_RECALIBRATE
BDA_TIMER_TO_CX_DX
                                                                            cmp
                                                                           je
call
                                                                           pop
                                                                                        bx
                                                                           bob
1505 OAA7 58
1506 0AA8 3B D9
1507 0AAA 77 E9
                                                                                        short DO TEST READY
                                                                            jа
                                                                           cmb
jp
1508 OAAC 72 04
                                                                                        short D0_TEST_READY_TIMED_OUT
1509 OAAE 3B C2
1510 OABO 77 E3
                                                                                        short D0_TEST_READY
                                                                           ja
1511
 1512 OAB2
                                                  DO_TEST_READY_TIMED_OUT:
 1512 0AB2 5A
                                                                                        dx
                                                                           pop
1514 OAB3 59
                                                                           pop
pop
1515 OAB4 5B
1516 OAB5 58
1517 OAB6 BE 1218 R
                                                                           pop
                                                                                        si. offset STRING C
1518 OAB9 EB 45
                                                                                        short DISK_INIT_ERROR
                                                                            jmp
 1519 OABB 90
                                                                           nop
1520
1521 OABC
                                                  DO_RECALIBRATE:
1522 OABC 5B
1523 OABD 58
                                                                           pop
                                                                           pop
pop
                                                                                        ax
dx
1524 OABE 5A
1525 OABF 59
1526 OACO 5B
                                                                           pop
                                                                           pop
mov
call
1527 OAC1 58
1528 OAC2 B4 11
1529 OAC4 E8 OB5A R
                                                                                        ah, INT13H_OP_11_RECALIBRATE
DO_CALL_INT13H
                                                                           jnc
mov
1530 OAC7 73 06
                                                                                        short DO_RESET_DISK_SYSTEM
1531 OAC9 BE 121A R
1532 OACC EB 32
                                                                                        si, offset STRING_D
short DISK_INIT_ERROR
                                                                            φmc
1533 OACE 90
1534
1535 OACF
                                                  DO_RESET_DISK_SYSTEM:
1536 OACF B4 00
1537 OAD1 E8 OB5A R
1538 OAD4 73 O6
1539 OAD6 BE 121C R
                                                                           mov
                                                                                        ah, INT13H_OP_00_RESET_DISK_SYSTEM
                                                                                        DO_CALL_INT13H
short DO_RESET_SUCCESSFUL
                                                                           call.
                                                                           jnc
                                                                                        si, offset STRING E
1540 OAD9 E8 OBOO R
                                                                                        DISK_INIT_ERROR
```

1541	0ADC		DO_RESET_SUCCES	SFUL:		
1542	OADC FE	06 0475 R		inc	BDA_NUMBER_OF_HARD_DISKS	
1543						
1544	OAEO B4	10		mov	ah, INT13H_OP_10_TEST_DRIVE_REAL	DY
1545	OAE2 E8	0B68 R		call	D1_CALL_INT13H	
1546	0AE5 73	08		jnc	short D1_RECALIBRATE	
1547	OAE7 80	FC 27		cmp	ah, INT13H_STATUS_27_NEED_RECAL	IBRATE
1548	0AEA 74	03		je	short D1_RECALIBRATE	
1549	OAEC EB	15		jmp	short ROM_INIT_DONE	
1550	OAEE 90			nop		
1551						
1552	OAEF		D1_RECALIBRATE:			
1553	OAEF B4	11		mov	ah, INT13H_OP_11_RECALIBRATE	
1554	OAF1 E8	0B68 R		call	D1_CALL_INT13H	
1555	0AF4 73	03		jnc	short D1_RECALIBRATE_SUCCESSFUL	
1556	OAF6 EB	0B		jmp	short ROM_INIT_DONE	
1557	OAF8 90			nop		
1558						
1559	OAF9		D1_RECALIBRATE_	SUCCESSF	UL:	
1560	OAF9 FE	06 0475 R		inc	BDA_NUMBER_OF_HARD_DISKS	
1561	OAFD EB	04		jmp	short ROM_INIT_DONE	
1562	OAFF 90			nop		
1563						
1564	0800		DISK_INIT_ERROR	:		
1565	0B00 E8	0B16 R		call	ERROR_1701	
1566	0B03		ROM_INIT_DONE:			
1567	0B03 E8	1139 R		call	ZERO_DS	
1568				ASSUME	ds: ZEROSEG	
1569	0B06 32	C0		xor	al, al	; Disable DMA/IRQ
1570	0B08 BA	0323		mov	dx, IO_PORT_323_DMA_IRQ	
1571	0B0B E8	103A R		call	DO_NOTHING	
1572	OBOE E8	1137 R		call	OUTB_DX_AL	
1573	0B11 B0	07		mov	al, 111b	; Mask DRQ3
1574	0B13 E6	OA.		out	OAh, al	
1575	0B15 CB			ret		
1576	0B16		ROM_INIT		ENDP	
1577						
1578						

```
1579
                                           PAGE
1580
1581
1582
                                                      This prints the "1701-?" error code when the option ROM fails to bring up the drive.
1583
1584
1585
                                                      The SI register points to -A, -B, \dots string which details the error encountered.
1586
1587
1588 OB16
1589 OB16 1E
                                           ERROR_1701
                                                                            PROC NEAR
                                                                push
                                                                           ds
1590 0B17 56
1591 0B18 BD 000F
                                                                push
mov
                                                                           si
bp, OFh
1592 OB1B 32 FF
                                                                xor
                                                                           bh, bh
1593 OBID BE 1210 R
1594 OB20 B9 0004
                                                                mov
                                                                           si, offset STRING_1701
cx, 4
1595 OB23 90
                                                                пор
1596 OB24 OE
1597 OB25 1F
                                                                 push
                                                                pop
Assume
1598
                                                                           ds:ROM
1599 0B26 E8 0B3C R
                                                                 call
                                                                           PRINT_CX_CHARS_FROM_DS_SI ; Print "1701"
1600 0B29 5E
1601 0B2A B9 0002
                                                                pop
1602 0B2D E8 0B3C R
                                                                 call
                                                                           PRINT_CX_CHARS_FROM_DS_SI ; Print (-A, -B, -C, -D, ...)
                                                                           si, offset STRING_CRLF
cx, 2
1603 0B30 BE 120E R
1604 0B33 B9 0002
                                                                 mov
                                                                nop
call
1605 OB36 90
1606 0B37 E8 0B3C R
                                                                           PRINT_CX_CHARS_FROM_DS_SI ; Print "\r\n"
1607 OB3A 1F
                                                                pop
                                                                           ds
1608 OB3B C3
                                                                 ret
1609 OB3C
                                           ERROR_1701
                                                                           ENDP
1610
1611
1612
1613
1614
                                                      A string print routine.
1615
1616 0B3C
1617 0B3C 50
                                           PRINT_CX_CHARS_FROM_DS_SI
                                                                                      PROC NEAR
                                                                push
                                                                           ax
si
1618 OB3D 56
                                                                 push
1619 OB3E
1620 OB3E AC
                                           NEXT_CHAR:
                                                                lodsb
1621 OB3F B4 OE
                                                                 mov
                                                                           ah, OEh
1622 0B41 CD 10
1623 0B43 49
                                                                 int
                                                                           10h
                                                                                                            ; TTY write
                                                                 dec
1624 OB44 75 F8
                                                                 jne
                                                                           short NEXT CHAR
1625 0B46 5E
1626 0B47 58
                                                                 pop
                                                                           ax
                                                                pop
1627 OB48 C3
1628 OB49
                                           PRINT_CX_CHARS_FROM_DS_SI
                                                                                      ENDP
1630
1631
1632
1633
                                                     Another string print routine. Unused.
1634
1635 OB49
1636 OB49 50
                                           UNUSED_PRINT_CX_CHARS_FROM_CS_SI
                                                                push
                                                                push
push
                                                                           si
ds
1637 OB4A 56
1638 OB4B 1E
1639 OB4C OE
                                                                 push
                                                                           CS
                                                                 pop
ASSUME
1640 OB4D 1F
                                                                           ds
1641
1642 OB4E
                                           UNUSED_NEXT_CHAR:
1643 OB4E AC
                                                                lodsb
                                                                           ah, 0Eh
10h
1644 OB4F B4 OE
1645 OB51 CD 10
1646 OB53 49
                                                                 mov
                                                                                                            ; TTY write
                                                                 dec
                                                                           CX
1647 0B54 75 F8
1648 0B56 1F
                                                                           short UNUSED_NEXT_CHAR ds
                                                                 pop ds
ASSUME ds:ZEROSEG
1649
1650 OB57 5E
1651 0B58 58
1652 0B59 C3
                                                                pop
1653 OB5A
                                           UNUSED_PRINT_CX_CHARS_FROM_CS_SI
                                                                                                 ENDP
1654
1655
1656
1657
1658
                                                      Issue a disk service routine with some
                                                     pre-set arguments for the first drive.
1659
1660
1661 OB5A
                                           DO_CALL_INT13H
                                                                           PROC NEAR
1662 0B5A 33 D2
1663 0B5C 81 CA 0080
1664 0B60 B9 0001
                                                                xor
                                                                           dx, dx
dx, 80h
                                                                                                ; DL = 80H = Drive U
; CH=0 = First Cylinder, CL=1 = First Sector
; AL=0 = Number of Sectors
                                                                or
                                                                           cx, 1
al, ⊪
1665 OB63 BO OO
                                                                mov
1666 0B65 CD 13
1667 0B67 C3
                                                                 int
                                                                           13h
                                                                ret
                                                                           ENDP
1668 OB68
                                           DO_CALL_INT13H
1669
1670
1671
1672
                                                     Issue a disk service routine with some pre-set arguments for the second drive
1674
1675
1676 OB68
1677 OB68 BA 0001
                                           D1_CALL_INT13H
                                                                           PROC NEAR
                                                                           dx, 1
dx, 80h
1678 OB6B 81 CA 0080
                                                                or
                                                                                                 : DL = 81H = Drive 1
1679 OB6F B9 0001
1680 OB72 B0 00
                                                                                                ; CH=0 = First Cylinder, CL=1 = First Sector
; AL=0 = Number of Sectors
                                                                 mov
                                                                 mov
1681 0B74 CD 13
                                                                int
                                                                           13h
1682 0B76 C3
1683 0B77
                                          D1_CALL_INT13H
                                                                           ENDP
1684
1685
```

```
1686
                                          PAGE
1687
1688
1689
                                                     BIOS Disk Services (INT 13H)
1690
1691 0B77
1692 0B77 FB
                                          INT13H_HANDLER
                                                                          PROC FAR
                                                                sti
1693 OB78 F6 C2 80
1694 OB7B 74 O2
1695 OB7D EB 05
                                                               test
                                                                          dl. 80h
                                                                          short NOT_A_HARD_DISK
short NOT_A_FLOPPY
                                                                jmp
1696 OB7F
                                          NOT_A_HARD_DISK:
1697 OB7F CD 40
                                                                int
                                                                          40h
                                                                                                          ; Invoke the old (floppy) ; INT13H handler
1698 OB81 EB 75
                                                                          short ALL_DONE
                                                                jmp
nop
1699 OB83 90
1700 OB84
                                          NOT_A_FLOPPY:
1700 0884
1701 0884 F6 C4 FF
1702 0887 75 04
1703 0889 50
                                                                          ah, OFFh
                                                                                                          ; Test for AH=00H ("Disk Subsystem Reset")
                                                                jле
                                                                          short HARD_DISK
                                                                                                          ; Handle a hard disk function
                                                                push
int
1704 0B8A CD 40
                                                                          40h
                                                                                                          ; Invoke the floppy reset handler and ; then also proceed with the hard disk
1705
1706 OB8C 58
                                                               pop
                                                                          ax
1707 OB8D
1708 OB8D 56
                                          HARD_DISK:
                                                               push
                                                                push
mov
1709 OB8E 51
1710 0B8F BE 121E R
1711 0B92 B9 0002
                                                                          si, offset INT13H_TRIVIAL_OP_VECTOR cx, 2 ; N
                                                                                                                    ; Number of vector entries
                                                                mov
1712 OB95
                                          NEXT TRIVIAL OF
1713 0B95 2E: 3A 24
                                                                          ah, cs:[si]
1714 0B98 74 0A
1715 0B9A 83 C6 03
                                                                          short TRIVIAL_OP_FOUND
                                                                je
add
                                                                          si. 3
                                                                                                                    ; Size of vector entry
1716 OB9D E2 F6
1717 OB9F 59
                                                                          NEXT_TRIVIAL_OP
                                                               pop
                                                               pop
jmp
nop
1718 OBAO 5E
1719 OBA1 EB 06
1720 OBA3 90
                                                                          short NO_TRIVIAL_OP_FOUND
1721 OBA4
                                          TRIVIAL_OP_FOUND:
1722 OBA4 59
1723 OBA5 2E: FF 64 01
1724 OBA9
                                                                          word ptr cs:[si+1]
                                                                jmp
                                          NO_TRIVIAL_OP_FOUND:
1725 OBA9 57
                                                                push
                                                                          di
1726 OBAA 56
1727 OBAB 06
                                                                push
                                                                push
                                                                          es
ds
1728 OBAC 1E
                                                                -
push
1729 OBAD 55
1730 OBAE 52
                                                               push
                                                                          bp
dx
                                                               push
1731 OBAF 51
                                                               push
push
                                                                          cx
1732 OBBO 53
1733 OBB1 80 FC 00
1734 OBB4 75 02
                                                                               INT13H_OP_00_RESET_DISK_SYSTEM
                                                                cmp
                                                                ine
                                                                          short PROCEED_HANDLING_FUNC
1735 OBB6 B2 80
1736
                                                                                                                    ; Floppy reset has been done ; Reset the hard disk now.
                                                                          dl, 80h
1737 OBB8
                                          PROCEED_HANDLING_FUNC:
1738 OBB8 E8 1139 R
                                                               call
ASSUME
                                                                          ZERO DS
                                                                          ds: ZEROSEG
1740 OBBB E8 ODFC R
                                                                          CHECK VALID DISK NUMBER
                                                                call
1741 OBBE 73 08
1742 OBCO C6 06 0474 R 01
1743 OBC5 EB OC
                                                                          SHORT DISK_NUMBER_IS_VALID
BDA_LAST_OP_STATUS, INT13H_STATUS_01_BAD_COMMAND
short CLEANUP
                                                                jnb
                                                                mov
                                                                jmp
                                                               nop
/ALID:
1744 OBC7 90
1745 OBC8
1746 OBC8 E8 OBFB R
                                                                          SET_BDA_STATUS_SUCCESS
                                                                call
                                                                          FILL_COMMAND_BUFFER
short CLEANUP
CONTROLLER_FUNC
1747 OBCB E8 OC06 R
                                                                call
1748 OBCE 72 03
1749 OBDO E8 OCBO R
                                                               jc
call
1750 OBD3
                                          CLEANUP:
1751 OBD3 50
1752 OBD4 E8 1139 R
                                                               push
call
                                                                          ax
ZERO_DS
1753
                                                               ASSUME
                                                                          ds: ZEROSEG
1754 OBD7 32 CO
1755 OBD9 BA 0323
1756 OBDC E8 103A R
                                                                nov
                                                                          al, al
dx, IO_PORT_323_DMA_IRQ
                                                                                                                    ; Disable DMA/IRQ
                                                                call
                                                                          DO NOTHING
1757 OBDF E8 1137 R
                                                               call
mov
                                                                          OUTB_DX_AL
al, 111b
                                                                                                                    ; Mask DRQ3
                                                                          OAh, al
1759 OBE4 E6 OA
                                                                out
1760 OBE6 58
1761 OBE7 8A 26 0474 R
                                                                          ah, BDA_LAST_OP_STATUS
1762 OBEB OA E4
                                                                or
                                                                          ah, ah
                                                                          short NO_ERROR
1763 OBED 74 01
1764 OBEF F9
1765 OBFO
                                          NO_ERROR:
1766 OBFO 5B
                                                               pop
pop
1767 OBF1 59
1768 OBF2 5A
                                                                          cx
dx
                                                               pop
1769 OBF3 5D
                                                               pop
1770 OBF4 1F
1771 OBF5 07
                                                               pop
                                                                          es
1772 OBF6 5E
                                                                          si
1773 OBF7 5F
                                                                          di
                                                               pop
1774 OBF8
                                          ALL_DONE:
1775 OBF8 CA 0002
                                                               ret
                                                                         2
1776 OBFB
1777
                                          INT13H_HANDLER
                                                                          ENDP
1778
1779
1780
                                                     Reset the last disk operation status
1781
1782
1783 OBFB
1784 OBFB 1E
                                          SET_BDA_STATUS_SUCCESS PROC NEAR
                                                               push
1785 OBFC E8 1139 R
                                                                call
                                                                         ZERO DS
1786
1787 OBFF C6 06 0474 R 00
                                                                          ds:ZEROSEG
BDA_LAST_OP_STATUS, INT13H_STATUS_00_NO_ERROR
                                                               mov
                                          pop ds ret
SET_BDA_STATUS_SUCCESS ENDP
1788 OC04 1F
1789 0C05 C3
1790 0C06
1791
1792
```

```
PAGE
1794
1795
1796
                                                           This takes the INT13H arguments from
                                                           registers and places them in command buffer
1797
                                                           for submission to the controller.
1798
1799
1800 0006
                                               FILL COMMAND BUFFER
                                                                                   PROC NEAR
1801 0C06 E8 0C7E R
1802 0C09 73 03
                                                                       call
                                                                                  VALIDATE_CHS
short CHS_VALID
                                                                       jnb
1803 OCOB E9 OCD1 R
                                                                       jmp
                                                                                   BAD_COMMAND_ERROR_RETURN
1804 OCOE
1805 OCOE 50
                                               CHS_VALID:
                                                                      push
1806 OCOF 53
                                                                      push
                                                                                  bx
1807 OC10 8A C4
1808 OC12 BB 122A R
                                                                                   al, ah
bx, offset INT13H_OP_TO_COMMAND_BYTE
1809 OC15 2E: D7
                                                                      xlat
                                                                                   byte ptr cs:[bx]
1810 OC17 A2 0442 R
1811 OC1A 5B
1812 OC1B 58
                                                                      mov
                                                                                   BDA_CONTROLLER_DATA_BUFFER_00, al
                                                                       pop
dec
1813 OC1C FE C9
1814 OC1E 89 OE 0444 R
1815 OC22 A2 0446 R
                                                                                   cl
                                                                                   word ptr BDA_CONTROLLER_DATA_BUFFER_02, cx
BDA_CONTROLLER_DATA_BUFFER_04, al
                                                                      mov
                                                                       mov
1816 OC25 80 FC OE
1817 OC28 74 05
1818 OC2A 80 FC OF
                                                                                   ah, OP_OE_READ_SECTOR_BUFFER
short READ_OR_WRITE
ah, OP_OF_WRITE_SECTOR_BUFFER
                                                                       cmp
1819 OC2D 75 05
                                                                                   short ANY OP
1820 OC2F
1821 OC2F C6 O6 0446 R 01
                                               READ_OR_WRITE:
                                                                                   BDA_CONTROLLER_DATA_BUFFER_04, 1
                                                                       mov
1822 OC34
                                               ANY_OP:
1822 0C34
1823 0C34 80 E6 1F
1824 0C37 88 36 0443 R
                                                                                   BDA_CONTROLLER_DATA_BUFFER_01, dh
1825 OC3B B1 03
                                                                      mov
                                                                                   c1, 3
1826 0C3D D2 CA
1827 0C3F 80 E2 60
1828 0C42 08 16 0443 R
                                                                       ror
and
                                                                                   dl, cl
dl, 60h
                                                                                   BDA_CONTROLLER_DATA_BUFFER_01, dl
                                                                       or
                                                                      push
push
1829 OC46 50
1830 0C47 57
1831 0C48 51
                                                                      push
                                                                                  CX
                                                                       push
call
1832 0049 06
                                                                                   95
1833 OC4A E8 1139 R
                                                                      ASSUME
1834
1835 OC4D E8 101E R
                                                                                  ds: ZEROSEG
                                                                       call
                                                                                  DRIVE_PARAM_TO_DI
1836 0C50 83 C7 08
1837 0C53 26: 8A 05
1838 0C56 A2 0447 R
                                                                       add
                                                                                  d1, # a1, ss:[di]
BDA_CONTROLLER_DATA_BUFFER_05, a1
BDA_CONTROLLER_DATA_BUFFER_00, OP_04_FORMAT_DRIVE
short FORMAT_DRIVE_OR_TRACK_NOT_BAD
                                                                       mov
                                                                      mov
                                                                       cmp
je
cmp
je
1839 OC59 80 3E 0442 R 04
1840 OC5E 74 OE
1841 0C60 80 3E 0442 R 06
1842 0C65 74 07
1843 0C67 80 3E 0442 R 07
1844 0C6C 75 0A
                                                                                   BDA_CONTROLLER_DATA_BUFFER_00, OP_06_FORMAT_TRACK
                                                                                  short FORMAT_DRIVE_OR_TRACK_NOT_BAD
BDA_CONTROLLER_DATA_BUFFER_00, OP_07_FORMAT_BAD_TRACK
short FORMAT_DRIVE_OR_TRACK
                                                                       ine
1845 OC6E
1846 OC6E 80 26 0444 R C0
1847 OC73 80 OE 0447 R OO
                                               FORMAT_DRIVE_OR
                                                                      TRACK
                                                                                  OT BAD:
                                                                                  BDA_CONTROLLER_DATA_BUFFER_02, 0C0h
BDA_CONTROLLER_DATA_BUFFER_05, 0
                                                                      and
                                                                       or
1848 0C78
1849 0C78 07
1850 0C79 59
                                               FORMAT_DRIVE_OR
                                                                      TRACK:
                                                                      pop
                                                                      pop
pop
1851 OC7A 5F
                                                                                   di
1852 0C7B 58
1853 0C7C F8
                                                                       clc
1854 OC7D C3
                                                                       ret
                                               FILL_COMMAND_BUFFER
1855 OC7E
                                                                                  ENDP
1856
1857
1858
1859
                                                           Checks that the coordinates are in bounds.
1860
1861
1862
                                                           Takes INT13H arguments sets carry on failure.
                                                                                      ents in CX/DX,
1863
1864
1865
                                                          Accepts C/H/S up to 614/4/17 Note that our drive actually is 615/2/34\ldots
1866
1867
                                               VALIDATE_CHS
1869 OC7E 50
                                                                      push
                                                                                   ax
                                                                      push
push
1870 OC7F 53
                                                                                  bx
1871 0C80 51
1872 0C81 52
                                                                       push
                                                                                   dж
1873 OC82 80 FA 01
1874 OC85 77 1F
1875 OC87 80 FE 03
                                                                       cmp
ja
                                                                                   dl. 1
                                                                                                          ; Check drive number
                                                                                   short BAD_ARGUMENT
dh, 3 ; Check head number
                                                                       cmp
                                                                                   dh, 3 ; (
short BAD_ARGUMENT
                                                                      ja
push
and
1876 OC8A 77 1A
1877 OC8C 51
1878 OC8D 80 E1 3E
                                                                                  cx
cl, 3Fh
1879 OC90 80 F9 11
                                                                       cmp
                                                                                   cl, 17
                                                                                                          : Check sector number
1880 0C93 59
1881 0C94 77 10
                                                                                   cx
short BAD_ARGUMENT
                                                                      pop
ja
and
1882 0C96 80 E1 C0
                                                                                   cl, 0C0h
1883 0C99 8B C1
1884 0C9B B1 02
1885 OC9D D2 C0
                                                                       rol
                                                                                   al, cl
                                                                       xchg
1886 OC9F 86 E0
1887 OCA1 3D 0265
1888 OCA4 76 04
                                                                       cmp
                                                                                                           ; Check cylinder
                                                                                   short ARGUMENTS_FINE
                                                                       jbe
1889 OCA6
                                               BAD_ARGUMENT:
1890 OCA6 F9
1891 OCA7 EB 02
                                                                                   short DONE
                                                                       jmp
1892 OCA9 90
1893 OCAA
1894 OCAA F8
                                               ARGUMENTS_FINE:
                                                                       clc
1895 OCAB
                                               DONE:
1896 OCAB 5A
1897 OCAC 59
                                                                      pop
                                                                                   CX
                                                                      pop
pop
ret
1898 OCAD 5B
                                                                                   bx
1899 OCAE 58
              С3
                                               VALIDATE_CHS
1901 OCB0
                                                                                  ENDP
1902
```

```
PAGE
1905
                                                                 This handles INT13H functions that require any sort of interaction with the controller hardware. This could be either a command submission, but also a reset.
1906
1907
1908
1909
1910
1911
1911
1912 OCBO
1913 OCBO 80 FC 14
1914 OCB3 77 1C
1915 OCB5 E8 OCD9 R
1916 OCB8 73 O6
                                                     CONTROLLER_FUNC
                                                                                            PROC NEAR
                                                                                             ah, LAST_INT13H_OP
                                                                               cmp
                                                                               ja
call
jnb
                                                                                             short BAD_COMMAND_ERROR_RETURN
                                                                                            CHECK_OP_NOT_A_RESET
short NOT_A_RESET
1917 OCBA E8 OCE7 R
                                                                                call
                                                                                            RESET_CONTROLLER
1918 OCBD EB 19
1919 OCBF 90
                                                                               jmp
nop
                                                                                             short JUST_RETURN
1920 OCC0
                                                    NOT_A_RESET:
1921 OCCO 80 FC 09
1922 OCC3 75 06
1923 OCC5 E8 OD83 R
                                                                               cmp
jne
call
                                                                                            ah, INT13H_OP_09_INITIALIZE_DISK_TABLE short RUN_COMMAND INITIALIZE_DISK_TABLE
                                                                               jmp
1924 OCC8 EB OE
                                                                                             short JUST_RETURN
1925 OCCA 90
1926 OCCB
                                                     RUN_COMMAND:
1927 OCCB E8 OEOC R
                                                                               call
                                                                                            DO COMMAND
1928 OCCE EB 08
1929 OCDO 90
                                                                                             short JUST_RETURN
                                                                                qmį
                                                                               пор
1930 OCD1
                                                     CONTROLLER FUNC
                                                                                            ENDP
1931
1932 OCD1
                                                     BAD_COMMAND_ERROR_RETURN
                                                                                                         PROC NEAR
1933 OCD1 C6 06 0474 R 01
                                                                                            BDA_LAST_OP_STATUS, INT13H_STATUS_01_BAD_COMMAND
1934 OCD6 F9
1935 OCD7 C3
                                                                                stc
1936 OCD8
                                                     BAD_COMMAND_ERROR_RETURN
                                                                                                         ENDP
1937
1938 OCD8
1939 OCD8 C3
                                                                               ret
1940 OCD9
                                                     JUST_RETURN
                                                                                            ENDP
1941
1942
1943
1944
1945
                                                                 Sets carry if the operation is a reset
1946
1947 OCD9
1948 OCD9 80 FC 00
                                                     CHECK_OP_NOT_A_RESET
                                                                                            PROC NEAR
ah, INT13H_OP_00_RESET_DISK_SYSTEM
                                                                               cmp
                                                                                            short Is_A_RESET
ah, INT13H_OP_OD_ALTERNATE_DISK_RESET
short Is_A_RESET
1949 OCDC 74 07
1950 OCDE 80 FC 0D
1951 OCE1 74 02
                                                                               je
cmp
je
clc
1952 OCE3 F8
1953 OCE4 C3
1954 OCE5
                                                     IS_A_RESET:
1955 OCE5 F9
                                                                               stc
1956 OCE6 C3
1957 OCE7
1958
1959
                                                                  This issues a reset signal to the controller
1961
                                                                 and then configures the drive geometry.

Done in response to INTI3H functions 00H

("Disk Subsystem Reset") and 0D ("Alternate
Disk Reset"), but also on errors in hope of
bringing the controller back to workable state.
1962
1963
1964
1965
1966
1967
1968
1969 OCE7
1970 OCE7 E8 OD61 R
                                                                                            PROC NEAR
MASTER_RESET
                                                                               call
1971 OCEA E8 OD83 R
                                                                               call.
                                                                                            INITIALIZE_DISK_TABLE
1972 OCED C3
1973 OCEE
                                                     RESET_CONTROLLER
1974
1975
1976
1977
1978
                                                                  Not sure why this exists or what command laH would do.
1980
1981 OCEE
                                                     UNUSED_1A
                                                                                            PROC NEAR
1981 OCEE
1982 OCEE C6 06 0442 R 1A
1983 OCF3 C7 06 0443 R 0000
1984 OCF9 C6 06 0445 R 06
1985 OCFE C6 06 0446 R 03
1986 0D03 C6 06 0447 R 00
                                                                                            BDA_CONTROLLER_DATA_BUFFER_00, 1Ah
word ptr BDA_CONTROLLER_DATA_BUFFER_01, 0000H
                                                                               mov
                                                                                            BDA_CONTROLLER_DATA_BUFFER_03, 06H
BDA_CONTROLLER_DATA_BUFFER_04, 03H
BDA_CONTROLLER_DATA_BUFFER_05, 00H
                                                                               mov
1987 ODOS ES OEOC R
                                                                               call
                                                                                            DO COMMAND
1988 ODOB C3
1989 ODOC
                                                     UNUSED_1A
1990
1991
```

```
1992
                                              PAGE
1993
1994
1995
                                                         This is the INT13H 01H ("Get Drive Parameters") handler. What it just chews and
1996
1997
1998
                                                          returns data that can be gotten straight from
                                                          the BDA anyways.
1999
2000 0D0C
2001 0D0C 5E
                                              INT13H_08_GET_DRIVE_PARAMS
                                                                                           PROC FAR
                                                                    pop
push
                                                                                si
di
2002 ODOD 57
2003 ODOE 1E
2004 ODOF 06
                                                                    push
push
2005 OD10 E8 1139 R
                                                                                ZERO_DS
                                                                     call
                                                                                ds: ZEROSEG
CHECK_VALID_DISK_NUMBER
2006
                                                                    ASSUME
2007 OD13 E8 ODFC R
                                                                     call
2008 0D16 73 0A
2009 0D18 C6 06 0474 R 01
2010 0D1D B4 01
2011 0D1F EB 2E
2012 0D21 90
                                                                     jnb
                                                                                short IS A VALID DISK NUMBER
                                                                                BDA_LAST_OP_STATUS, INT13H_STATUS_01_BAD_COMMAND
                                                                                short DONE_GETTING_PARAMS
                                              jmp
nop
IS_A_VALID_DISK_NUMBER
2013 0D22
2014 0D22 E8 0BFB R
                                                                                SET_BDA_STATUS_SUCCESS
                                                                    call
                                                                                cl, 3
dl, cl
dl, 60h
2015 0D25 B1 03
2015 0D25 B1 03
2016 0D27 D2 CA
2017 0D29 80 E2 60
2018 0D2C 88 16 0443 R
2019 0D30 E8 101E R
2020 0D33 8A 16 0475 R
2021 0D37 26: 8A 75 02
                                                                     and
                                                                    mov
                                                                                BDA_CONTROLLER_DATA_BUFFER_01, dl
                                                                                DRIVE_PARAM_TO_DI
dl, BDA_NUMBER_OF_HARD_DISKS
                                                                     call
                                                                    mov
                                                                    mov
                                                                                dh, es:[di+2]
2022 OD3B FE CE
2023 OD3D 26: 8B 05
2024 OD40 48
                                                                    dec
mov
                                                                                ax, es:[di]
                                                                     dec
                                                                                ax
2025 0D41 48
2026 0D42 8A E8
2027 0D44 B1 06
                                                                    dec
mov
                                                                                ax
ch, al
cl, 6
                                                                    mov
2028 OD46 D2 E4
2029 OD48 86 E1
2030 OD4A 80 C9 11
                                                                                ah, cl
ah, cl
                                                                     sh1
                                                                     xchg
                                                                     or
                                                                                cl, 11h
2031 0D4D 33 CO
                                                                     xor
2032 0D4F
2033 0D4F 07
                                              DONE_GETTING_PA
                                                                    RAMS:
                                                                    pop
                                                                                es
ds
2034 OD50 1F
2035 0D51 5F
2036 0D52 CA 0002
                                                                                di
                                                                     pop
                                              INT13H 08 GET DRIVE PARAMS
2037 OD55
                                                                                           ENDP ; sp =
2038
2040
2041
2042
                                                       Not sure why this exists.
2043
2044 0D55
2045 0D55 B1 03
2046 0D57 D2 CA
                                              UNUSED_BYTE_01
                                                                                PROC NEAR
                                                                                cl, 3
dl, cl
                                                                    ror
2047 0D59 80 E2 60
2048 0D5C 88 16 0443 R
2049 0D60 C3
                                                                     and
                                                                                dl. 60h
                                                                                BDA_CONTROLLER_DATA_BUFFER_01, dl
                                                                    ret
2050 OD61
                                              UNUSED_BYTE_01
                                                                                ENDP
2051
2052
2053
2054
2055
                                                       This issues a reset signal to the controller. :
2056
2057 0D61
2058 0D61 BA 0321
                                              MASTER_RESET
                                                                                PROC NEAR
                                                                                dx, IO_PORT_321_READ_STATUS_WRITE_RESET
2059 OD64 E8 103A R
2060 OD67 E8 1137 R
2061 OD6A B9 0684
                                                                    call
                                                                                DO NOTHING
                                                                    call
mov
                                                                                OUTB_DX_AL
cx, 684h
                                                                                                                 ; Arbitrary delay
2062 OD6D
                                             DO_LOOP:
                                                                    loop
2063 0D6D E2 FE
                                                                                DO_LOOP
2065 OD70
                                              MASTER_RESET
                                                                                ENDP
2066
2068
                                                         This is the INT13H 01H ("Get Disk Status") handler. It returns status of the last operation in AL and clears it.
2069
2070
2071
2072
2073
2074 0D70
                                              INT13H_01_GET_DISK_STATUS
                                                                                           PROC FAR
                                                                               si
ah, ah
ds
2075 0D70 5E
                                                                    pop
2076 0D71 32 E4
2077 0D73 1E
                                                                    push
2078 0D74 E8 1139 R
                                                                                ZERO DS
                                                                     call
2079
                                                                    ASSUME ds:ZEROSEG
2080 OD77 AO 0474 R
                                                                                al, BDA_LAST_OP_STATUS
2081 OD7A 88 26 0474 R
                                                                                BDA_LAST_OP_STATUS, ah
                                                                    mov
2082 OD7E 1F
2083 OD7F F8
2084 0D80 CA 0002
                                                                     ret
2085 OD83
                                              INT13H_01_GET_DISK_STATUS
                                                                                           ENDP
2087
```

```
2088
                                         PAGE
2089
2090
                                                   Load disk geometry to the controller.
2091
                                                   This is done in response to INT13H op 09H ("Initialize Disk Table"), but also on reset and when dealing with error conditions.
2092
2094
2095
                                                   Note that this forces the head count ts 2 regardless of what's in the INT14H table; which happens to have head count of 4. Not
2096
2097
2098
2099
                                                   sure why and the controller seems to happily accepts heads over 2 anyways.
2101
2102
2103
                                         INITIALIZE_DISK_TABLE
                                                                       PROC NEAR
2104 0D83 C6 06 0442 R 0C
2105 0D88 80 26 0443 R 00
                                                             mov
                                                                        BDA CONTROLLER DATA BUFFER 00, OP 0C_INIT_DRV PARM
                                                                                                                          ; Start with the ; first drive
                                                              and
                                                                        BDA_CONTROLLER_DATA_BUFFER_01, 00h
2107
2108 OD8D
                                         START_INIT_DRV_PARM_COMMAND:
                                                                       AND:
SELECT_BUSY_FOR_COMMAND
WRITE_COMMAND_BUFFER
short COMMAND_ISSUED_SUCCESS
2109 OD8D E8 OF39 R
2110 OD90 E8 OEFC R
                                                             call
2111 OD93 73 O9
                                                              jnb
2112
2113 OD95
                                         BAD_TABLE_ERROR_RETURN:
2114 0D95 E8 0D61 R
                                                             call
                                                                       MASTER RESET
2115 OD98 C6 O6 O474 R O7
2116 OD9D C3
                                                                       BDA_LAST_OP_STATUS, INT13H_STATUS_07_BAD_DISK_PARAM_TABLE
                                                             ret
2117
2118 OD9E
                                         COMMAND_ISSUED_SUCCESS:
2119 OD9E E8 OF45 R
                                                                       WAIT_FOR_BYTE_READY
                                                             call
                                                                        short PROCEED_WRITING_DRIVE_PARAMS
2120 ODA1 73 02
                                                              jnb
2121 ODA3 EB FO
                                                                        short BAD_TABLE_ERROR_RETURN
                                                              jmp
2123 ODA5
                                         PROCEED_WRITING_DRIVE_PARAMS:
                                                             push
call
2124 ODA5 06
2125 ODA6 E8 1139 R
                                                                        ZERO_DS
2126
                                                             ASSUME
                                                                       ds: ZEROSEG
2127 ODA9 ES 101E R
                                                             call.
                                                                       DRIVE PARAM TO DI
2128 ODAC 8B DF
2129 ODAE B9 0008
                                                                        bx, di
                                                             mov
                                                                                                                           ; Eight drive params bytes
                                                             mov
                                                                        cx, ■
2130 ODB1 33 F6
                                                              xor
                                                                        si, si
2131
2132 ODB3
                                         WRITE_DRIVE_PARAM_BYTE:
2133 ODB3 E8 OF45 R
                                                             call
                                                                       WAIT FOR BYTE READY
2134 ODB6 56
2135 ODB7 53
                                                             push
                                                             push
2136 ODB8 BB ODF4 R
                                                              mov
                                                                       bx, offset DATA_FROM_INT41H_OFFSETS
2137 ODBB 2E: 8A 18
2138 ODBE 32 FF
                                                              MON
                                                                       bl, cs:[bx+si]
bh, bh
                                                                                                                           ; Read offset from the table
2139 ODCO 8B F3
                                                             mov
                                                                        si, bx
2140 ODC2 5B
2141 ODC3 26: 8A 00
2142 ODC6 83 FE 02
                                                             pop
                                                                       al, es:[bx+si]
si, 2
                                                                                                                             Read data from drive table
                                                             jne
                                                                                                                           : Offset 2 is head number
2142 ODCS 83 FE
2143 ODC9 75 02
2144 ODCB B0 02
2145
                                                                       short OFFSET_NOT_TWO
al, 2
                                                                                                                             Force head number to two.
2146 ODCD
2147 ODCD BA 0320
2148 ODDO E8 103A R
2149 ODD3 E8 1137 R
                                         OFFSET_NOT_TWO:
                                                                       dx, IO_PORT_320_DATA
DO_NOTHING
                                                              call
                                                              call
                                                                        OUTB_DX_AL
2150 ODD6 5E
                                                              pop
2151 ODD7 46
2152 ODD8 E0 D9
                                                              inc
                                                             loopne
                                                                       WRITE_DRIVE_PARAM_BYTE
2153 ODDA 07
2154 ODDB E8 OF6F R
                                                             pop
call
                                                             cmp
jne
                                                                        BDA_LAST_OP_STATUS, INT13H_STATUS_00_NO_ERROR
2155 ODDE 80 3E 0474 R 00
2156 ODE3 75 OE
                                                                        short DONE_INITIALIZING_DISK_TABLE
2157
2158 ODE5 F6 O6 0443 R 20
2159 ODEA 75 O7
                                                             test
                                                                       BDA CONTROLLER DATA BUFFER 01, 20h
                                                                                                                           ; Were we configuring
                                                                        short DONE_INITIALIZING_DISK_TABLE
                                                                                                                              the second drive?
2161 ODEC C6 06 0443 R 20
                                                                       BDA_CONTROLLER_DATA_BUFFER_01, 20h
                                                             mov
                                                                                                                           ; Proceed with the
2162 ODF1 EB 9A
                                                              jmp
                                                                        short START_INIT_DRV_PARM_COMMAND
                                                                                                                              second drive.
2164 ODF3
                                        DONE_INITIALIZING DISK_TABLE:
2165 ODF3 C3
2166 ODF4
2167
                                         INITIALIZE_DISK_TABLE ENDP
2168
2169
2170
                                                   Mapping from INT41H format to what the
2171
                                                   controller command OCH ("Init drive
                                                   parameters") expects. It essientially just flips endianness of WORD values.
2172
2173
2174
2175
2176 ODF4
                                         DATA_FROM_INT41H_OFFSETS:
2177 ODF4 01
                                                             DB
                                                                                            ; Max cyls Hi
2178 ODF5 00
2179 ODF6 02
                                                             DB
DB
                                                                                              Max cyls Lo
Max heads
                                                                                            ; RPC Hi
2180 ODF7 04
                                                             DB
2181 ODES 03
                                                             DB
                                                                                              RPC To
2183 ODFA 05
                                                                                            ; WPC Lo
2184 ODFB 07
                                                                                            : ECC Len
2185
2186
```

```
2187
                                          PAGE
2188
2189
2190
                                                     Sets carry if not a drive 80H or 81H
2191
2192 ODFC
2193 ODFC 80 EA 80
                                           CHECK_VALID_DISK_NUMBER PROC NEAR
                                                                sub
                                                                          dl, 80h
dl, 2
                                                               cmc
2194 ODFF 80 FA 02
2194 ODFF 80
2195 OEO2 F5
2196 OEO3 C3
2197 OEO4
                                          CHECK_VALID_DISK_NUMBER ENDP
2198
2199
2200
2201
                                                     Not sure why this exists...
2202
2203
2204 OE04
                                          UNUSED_1B
                                                                          PROC NEAR
2205 0E04 C6 06 0442 R 1B
2206 0E09 EB 01
                                                                           BDA_CONTROLLER_DATA_BUFFER_00, 1Bh
                                                               jmp
nop
                                                                          short DO_COMMAND
2207 OEOB 90
2208 0E0C
                                          UNUSED_1B
2209
2210
2211
2212
                                                    This is the command submission routine
It sets up DMA if necessary, submits the
command, reads the response and deals with
the error sense if necessary.
2213
2214
2215
2216
2217
2218 0E0C
                                          DO_COMMAND
                                                                          PROC NEAR
2219 0E0C E8 0E9E R
                                                                          CHECK_COMMAND_USES_DMA
                                                               call
2220 0E0F 73 68
2221 0E11 06
                                                               jnc
push
                                                                           short ISSUE_THE_COMMAND ; Just issue it, no DMA
2222 OE12 58
                                                                pop
                                                                          ax
2223 OE13 B1 04
                                                                          cl, 4
2224 0E15 D3 E0
2225 0E17 03 C3
                                                                shl
                                                                add
                                                                          ax, bx
2226 OE19 B9 FFFF
                                                                mov
                                                                          cx. OFFFFh
                                                                                                          : Maximum ISA DMA length -- 64K
2227 OE1C 2B C8
2228 OE1E 91
                                                                           сж, аж
                                                                xchq
                                                                          ax, cx
2229 0E1F E8 0ED2 R
                                                                call
                                                                          SET DATA LENGTH
2230 0E22 3B C1
2231 0E24 73 06
                                                                          ax, cx
short DMA_ADDRESS_FINE
                                                                jnb
2232 OE26
                                          BAD DMA:
2233 0E26 C6 06 0474 R 09
2234 0E2B C3
                                                               mov
ret
                                                                          BDA_LAST_OP_STATUS, INT13H_STATUS_09_DMA_ACROSS_64K
2235
2236 0E2C
2237 0E2C B0 80
2238 0E2E E8 0EEA R
                                          DMA_ADDRESS_FINE:
                                                                           al, 128
                                                                          NOT A LONG COMMAND
                                                                call
2239 0E31 73 02
2240 0E33 B0 7F
                                                                jnb
                                                                          short CONFIGURE_DMA
al, 127
                                                                mov
2241 0E35
                                          CONFIGURE DMA:
2242 0E35 38 06 0446 R
2243 0E39 77 EB
2244
                                                               cmp
ja
                                                                          BDA_CONTROLLER_DATA_BUFFER_04, al
2245 0E3B FA
                                                                cli
2246 0E3C E8 0EBD R
2247 0E3F E6 0B
                                                                call
                                                                          SELECT_DMA_MODE
                                                                                                         ; IN or OUT
; Set DMA mode
                                                                out
                                                                           OBh, al
                                                                jmp
out
2248 0E41 EB 00
                                                                           short $+2
2249 0E43 E6 0C
2250
                                                                                                          ; Clear DMA counter
                                                                           OCh, al
2251 0E45 06
                                                               push
                                                                          es
2252 0E46 58
2253 0E47 8B C8
                                                               pop
                                                                          cx,
                                                                          cx, ax cl, 4
2254 0E49 B1 04
                                                               mov
2255 0E4B D2 ED
2256 0E4D D3 E0
                                                                shr
shl
                                                                           ax,
2257 0E4F 03 C3
                                                                add
                                                                          ax,
                                                                               bx
2258 0E51 80 D5 00
2259 0E54 BA 0006
                                                               adc
mov
                                                                          ch,
dx,
2260 0E57 EE
                                                                                                          ; DMA base bits 0-7
                                                                out
                                                                          dx,
                                                                               al
2261 0E58 86 C4
                                                                xchg
                                                                           al, ah
2262 0E5A EE
2263 0E5B 86 C5
                                                                          dx, al
al, ch
                                                                out
                                                                                                          ; DMA base bits 8-15
                                                               xchq
                                                                                                          ; Page register
; DMA base bits 16-23
2264 0E5D BA 0082
                                                                           dx, 82h
2265 0E60 EE
2266
2267 0E61 E8 0ED2 R
                                                               call
                                                                          SET DATA LENGTH
2268 0E64 8B C1
2269 0E66 BA 0007
                                                                          ax, cx
dx, 7
                                                               mov
2270 OE69 EE
                                                                out
                                                                          dx. al
                                                                                                          : Word count low
2271 0E6A 86 E0
2272 0E6C EE
                                                                xchg
                                                                           ah, al
                                                                                                          ; Word count high
                                                                out
                                                                           dx, al
2273
2274 0E6D FB
                                                                sti
2275 0E6E B0 01
2276 0E70 BA 0323
                                                                                                          ; IRQEN=0 DMAEN=1 -- Enable DMA
                                                                mov
                                                                          dx, IO_PORT_323_DMA_IRO
                                                               mov
                                                                          DO_NOTHING
OUTB_DX_AL
2277 0E73 E8 103A R
2278 0E76 E8 1137 R
                                                                call
2279
2280 OE79
                                           ISSUE_THE_COMMAND:
2281 0E79 E8 0F39 R
2282 0E7C E8 0EFC R
                                                                          SELECT_BUSY_FOR_COMMAND
WRITE_COMMAND_BUFFER
                                                                call
                                                               pushf
and
2283 0E7F 9C
2284 0E80 80 26 0447 R 3F
2285 0E85 9D
                                                                           BDA_CONTROLLER_DATA_BUFFER_05, 3Fh
                                                               popf
2286 0E86 73 09
                                                                jnb
                                                                           short COMMAND_REQUIRES_RESPONSE
2287 0E88 E8 0D83 R
2288 0E8B C6 06 0474 R 80
                                                                          INITIALIZE_DISK_TABLE
BDA_LAST_OP_STATUS, INT13H_STATUS_80_TIMEOUT
                                                                mov
2289 0E90 C3
                                                                ret
2290 0E91
2291 0E91 E8 0E9E R
2292 0E94 73 04
                                          COMMAND_REQUIRES_RESPONSE:
                                                                          CHECK_COMMAND_USES_DMA
                                                                jnb
                                                                          short READ_RESPONSE
2293 0E96 B0 03
2294 0E98 E6 0A
                                                                           al, 011b
                                                                                                         ; Unmask DRQ3
                                                               out
2295 0E9A
                                          READ_RESPONSE:
2296 0E9A E8 0F6F R
                                                               call
                                                                          HANDLE_COMMAND_RESPONSE
                                          DO_COMMAND
2298 OE9E
                                                                          ENDP
2299
```

```
2301
                                         PAGE
2302
2303
2304
                                                   Sets carry if the command requires a DMA transfer for the data
2305
2306
2307 0E9E
                                         CHECK_COMMAND_USES_DMA
                                                                       PROC NEAR
2308 0E9E 56
2309 0E9F 51
2310 0EA0 50
                                                             push
push
                                                                        si
                                                              push
2311 OEA1 BE 1224 R
                                                              mov
                                                                        si, offset DMA_COMMAND_TABLE
2312 0EA4 B9 0006
2313 0EA7 90
                                                              пор
2314 OEA8 AO 0442 R
                                                                        al, BDA_CONTROLLER_DATA_BUFFER_00
                                                              mov
2315 OEAR
                                                      COMMAND TABLE
2315 OEAB
2316 OEAB 2E: 3A 04
2317 OEAE 74 08
2318 OEBO 46
                                                              cmp
                                                                        short COMMAND_USES_DMA
                                                              je
                                                              inc
loop
2319 OEB1 E2 F8
2320 OEB3 58
                                                                        CHECK_DMA_COMMAND_TABLE_ENTRY
                                                              pop
2321 OEB4 59
2322 OEB5 5E
2323 OEB6 F8
                                                              pop
clc
2324 0EB7 C3
                                                              ret
2324 OEB7 CS
2325 OEB8
2326 OEB8 58
2327 OEB9 59
                                         COMMAND_USES_DMA
                                                              pop
                                                                        ax
                                                              pop
pop
                                                                        cx
si
2328 OEBA 5E
2329 OEBB F9
                                                              stc
2330 OEBC C3
                                                              ret
2331 0EBD
2332
2333
2334
2335
                                                   Sets AL to DMA mode byte appropriate for
2336
                                                   given command, depending on the direction
                                                    of the transfer necessary.
2337
2339
2340 OEBD
                                         SELECT DMA MODE
                                                                        PROC NEAR
2341 OEBD AO 0442 R
2342 OECO 3C 08
2343 OEC2 74 OB
                                                                        al, BDA_CONTROLLER_DATA_BUFFER_00
al, OP_08_READ_SECTORS
                                                              cmp
                                                              je
cmp
                                                                         short READ COMMAND
2344 0EC4 3C 0E
2345 0EC6 74 07
                                                                        al, OP_0E_READ_SECTOR_BUFFER
short READ_COMMAND
                                                              je
cmp
2346 OEC8 3C E5
                                                                        al, OP E5 READ LONG
2347 OECA 74 03
2348 OECC
                                                                         short READ_COMMAND
                                         WRITE_COMMAND:
2349 OECC BO 4B
                                                              mov
                                                                        al, DRO3_WRITE
2350 OECE C3
2352 OECF BO 47
                                                                        al, DRQ3_READ
                                                              mov
2353 OED1 C3
2354 OED2
                                         SELECT_DMA_MODE
2355
2356
2357
2358
                                                   Sets CX to 518 if the command is
2359
                                                   Read/Write Long, 512 otherwise.
2360
2361
2362 OED2
                                         SET_DATA_LENGTH
                                                                        PROC NEAR
2363 OED2 50
2364 OED3 B9 0206
                                                              push
                                                                        ax
cx, 518
                                                                                                       ; Long command length: 512 + E (ecc?)
2365 OED6 E8 OEEA R
                                                              call
                                                                        NOT A LONG COMMAND
2366 OED9 72 03
2367 OEDB B9 0200
                                                                        short DATA_LENGTH_SET cx, 512
                                                              jc
                                                                                                       ; Regular command length
                                                              mov
2368 OEDE
                                         DATA_LENGTH_SET:
2369 OEDE 32 E4
2370 OEEO AO 0446 R
                                                                        ah, ah
al, BDA_CONTROLLER_DATA_BUFFER_04
2371 OEE3 F7 E1
2372 OEE5 48
2373 OEE6 8B C8
                                                              mul
                                                                        ÇX
                                                              dec
                                                                        ax
cx,
2374 OEE8 58
                                                              pop
ret
                                                                        ax
2375 OEE9 C3
2376 OEEA
                                         SET_DATA_LENGTH
2377
2378
2379
2380
                                                   Sets carry if the command is
2381
                                                   Read Long or Write Long
2382
2383
2384 OEEA
                                                LONG COMMAND
                                                                        PROC NEAR
2385 OEEA 80 3E 0442 R E5
2386 OEEF 75 02
                                                                        BDA_CONTROLLER_DATA_BUFFER_00, OP_E5_READ_LONG
                                                                        short NOT_READ_LONG
                                                              јле
2387 OEF1
                                         READ LONG OR WRITE LONG:
2388 0EF1 F9
2389 0EF2 C3
                                                              ret
2390 OEF3
                                         NOT_READ_LONG:
2391 OEF3 80 3E 0442 R E6
2392 OEF8 74 F7
                                                                        BDA_CONTROLLER_DATA_BUFFER_00, OP_E6_WRITE_LONG short READ_LONG_OR_WRITE_LONG
                                                              je
clc
2393 OEFA F8
2394 OEFB C3
2395 OEFC
                                         NOT_A_LONG_COMMAND
                                                                        ENDP
2396
2397
```

```
2398
                                          PAGE
2399
2400
2401
                                                    Write the while 6-byte command buffer Sets carry on failure (timeout)
2402
2403
2404 OEFC
                                          WRITE_COMMAND_BUFFER
                                                                          PROC NEAR
2405 0EFC E8 0F19 R
2406 0EFF 73 01
2407 0F01 C3
                                                                         WAIT_READY_FOR_COMMAND
short READY_WAIT_SUCCESSFUL
                                                               call
                                                               jnb
                                                               ret
2408 OF02
                                          READY_WAIT_SUCCESSFUL:
2409 OF02 BE 0442 R
2410 OF05 33 C9
2411 OF07 BA 0320
                                                               mov
                                                                          si, offset BDA_CONTROLLER_DATA_BUFFER_00
                                                                          cx, cx
dx, IO_PORT_320_DATA
                                                               mov
2412 OFOA E8 103A R
                                                               call
                                                                          DO_NOTHING
2413 OFOD
                                          WRITE_COMMAND_BYTE:
2414 OFOD 8A 04
                                                               mov
                                                                          al, [si]
2415 OFOF EE
2416 OF10 46
2417 OF11 FE C1
                                                               out
                                                               inc
                                                               inc
                                                                          cl
2417 OF11 FE C1
2418 OF13 80 F9 06
2419 OF16 75 F5
2420 OF18 C3
                                                               jne
                                                                          cl. 6
                                                                          short WRITE_COMMAND_BYTE
                                                               ret
2421 OF19
                                          WRITE_COMMAND_BUFFER
                                                                         ENDP
2422
2423
2424
2425
                                                    This waits until the controller is ready
2426
                                                    to accept a command byte.
2427
                                                    Sets carry on failure (timeout)
2428
2429
2430
2431 0F19
2432 0F19 B9 FFFF
2433 0F1C BA 0321
                                          WAIT_READY_FOR_COMMAND
                                                                         PROC NEAR
                                                                         cx, Offffh ; Try Oxffff times dx, IO_PORT_321_READ_STATUS_WRITE_RESET
                                                               mov
2434 OF1F E8 103A R
2435 OF22
2436 OF22 E8 1135 R
                                                               call
                                                                          DO_NOTHING
                                          SEE_IF_READY:
                                                               call
                                                                         INB_AL_DX
2437 OF25 24 OD
                                                               and
                                                                          al, 1101b
                                                                                                         : BUSY=1 COMMANDO/DATA1=1
2438
2439 OF27 3C OD
                                                                                                         ; INPUT1/OUTPUT0=0 REQ_READY=1
; The controller is ready to
                                                                         al, 1101b
                                                               cmp
2440
                                                                                                             accept a command byte.
2441 0F29 75 02
2442 0F2B F8
                                                                          short NOT_READY_YET
                                                               clc
2443 OF2C C3
                                                               ret
2444 0F2D
2445 0F2D E2 F3
                                          NOT_READY_YET:
                                                               loop
                                                                          SEE_IF_READY
2446 0F2F C6 06 0474 R 80
2447 0F34 E8 0D61 R
2448 0F37 F9
2449 0F38 C3
                                                               mov
                                                                          BDA LAST OP STATUS, INT13H STATUS 80 TIMEOUT
                                                               call
                                                                          MASTER_RESET
                                                               ret
2450 OF39
                                          WAIT_READY_FOR_COMMAND ENDP
2451
2452
2453
2454
2455
                                                    Issue a select signal.
This has to be done to start a command.
2456
                                                     The controller responds by enabling BUSY
2457
2458
                                                    in status byte.
2459
2460 OF39
2461 OF39 BA 0322
                                          SELECT_BUSY_FOR_COMMAND PROC NEAR
                                                                         dx, IO_PORT_322_READ_CONFIG_WRITE_SELECT
DO_NOTHING
                                                               mov
call
2462 OF3C E8 103A R
2463 OF3F B0 O1
2464 OF41 E8 1137 R
                                                               mov
call
                                                                         al, 1
OUTB_DX_AL
                                                                                                         ; Value is ignored
2465 OF44 C3
                                                               ret
2466 OF45
                                          SELECT_BUSY_FOR_COMMAND ENDP
2468
2469
2470
                                                    Wait until a byte can be read
2471
                                                    from the contoller
2472
2473
2474
                                                    Sets carry on failure (timeout)
2475
2476 OF45
2477 OF45 51
                                                               push
                                                                          CX
2478 OF46 52
                                                               -
push
                                                                          dx
2479 0F47 50
2480 0F48 B9 001E
                                                                          cx, 30
                                                                                                        ; 30 x 0xfffff times...
                                                               mov
2481
2482 OF4B
2483 OF4B 51
2484 OF4C B9 FFFF
                                          OUTER_CHECK_READY:
                                                               push
                                                                          cx. OFFFFh
                                                                                                          : ...30 x 0xffff times
                                                               mov
2485 0F4F BA 0321
2486 0F52 E8 103A R
                                                                         dx, io_PORT_321_READ_STATUS_WRITE_RESET
DO_NOTHING
                                                               call
2487 OF55
                                          INNER_CHECK_READY:
2488 OF55 E8 1135 R
                                                               call
and
                                                                          INB AL DX
2489 OF58 24 01
2490 OF5A 75 OE
                                                                                                         ; REQUEST=1
                                                                          al, 1
                                                                          short SUCCESSFULLY_READY
                                                               jnz
2491 OF5C E2 F7
                                                                          INNER_CHECK_READY
2491 OF5E E2 F7
2492 OF5E 59
2493 OF5F E2 EA
                                                               pop
                                                                         OUTER_CHECK_READY
                                                               loop
2494
2495 0F61 C6 06 0474 R 80
2496 0F66 F9
                                                                          BDA_LAST_OP_STATUS, INT13H_STATUS_80_TIMEOUT
                                                               stc
                                                               jmp
nop
2497 OF67 EB 02
                                                                          short DONE WAITING
2498 OF69 90
2499
2500 OF 6A
                                          SUCCESSFULLY READY:
2501 OF6A 59
2502 OF6B
                                                               pop
2503 OF6B 58
                                                               pop
pop
2504 OF6C 5A
2505 OF6D 59
                                                                          dx
                                                                          СЖ
                                                               pop
2506 OF6E C3
2507 OF6F
                                          WAIT_FOR_BYTE_READY
                                                                         ENDP
```

2509

```
PAGE
2511
                                                        Wait for command to finish with an appropriate timeout. Then check status byte and possibly handle an error. Disable DMA on disk and DMA controllers afterwards.
2512
2513
2514
2515
2516
2517
2518 OF6F
2519 OF6F 06
                                             HANDLE_COMMAND_RESPONSE PROC NEAR
                                                                   push
call
                                                                               es
ZERO_DS
2520 OF70 E8 1139 R
2521
2522 OF73 E8 101E R
                                                                               DRIVE_PARAM_TO_DI
                                                                    call
2522 OF73 E8 101E R
2523 OF76 26: 8A 4D 0A
2524 OF7A 80 3E 0442 R 04
2525 OF7F 74 OF
2526 OF81 26: 8A 4D 0B
2527 OF85 80 3E 0442 R E3
2528 OF8A 74 04
2529 OF8C 26: 8A 4D 09
                                                                               DRIVE_PARAME_TO_DI

cl, es:[di+(TIMEOUT_FMT - INT41H_DATA)]

BDA_CONTROLLER_DATA_BUFFER_00, OP_04_FORMAT_DRIVE

short CL_TIMEOUT_SET
                                                                    mov
                                                                    cmp
je
                                                                               cl, es: [di+(TIMEOUT CHK - INT41H DATA)]
                                                                    mov
                                                                    cmp
je
mov
                                                                               BDA_CONTROLLER_DATA_BUFFER_00, OP_E3_DRIVE_DIAG
short CL_TIMEOUT_SET
cl, es:[di+(TIMEOUT_STD - INT41H_DATA)]
2530 OF90
                                             CL_TIMEOUT_SET:
2531 0F90 07
2532
                                                                    pop
2533 OF91 32 ED
                                                                    xor
                                                                               ch, ch
                                                                                ax, 444h
2534 0F93 B8 0444
2535 0F96 F7 E1
                                                                    mul
                                                                    push
call
2536 OF98 52
2537 OF99 E8 1028 R
                                                                               BDA_TIMER_TO_CX_DX
2538 OF9C 03 D0
                                                                    add
                                                                               dx, ax
2539 OF9E 83 D1 00
                                                                    adc
2540 OFA1 5B
2541 OFA2 03 CB
2542 OFA4 8B D9
                                                                    mov
                                                                               bx, cx
2543 OFA6 8B C2
2544
2545 OFA8
                                             CHECK_BYTE_READY_FOR_R
                                                                              EAD:
2546 OFA8 50
2547 OFA9 53
2548 OFAA BA 0321
                                                                    push
                                                                               dx, IO_PORT_321_READ_STATUS_WRITE_RESET
2549 OFAD E8 103A R
                                                                    call.
                                                                               DO NOTHING
2550 OFBO E8 1135 R
2551 OFB3 24 07
                                                                               al, 111b
                                                                                                                            ; BUSY=0 CONTROLO/DATA1=1
                                                                    and
2552
                                                                                                                                INPUT1/OUTPUT0=1 REO BYTE=1
2553 OFB5 3C 07
2554
                                                                                                                                Is data byte ready for input to host?
                                                                               al, 111b
                                                                    cmp
2555 OFB7 75 05
                                                                    ine
                                                                               short DATA_BYTE_NOT_READY
2556 OFB9 5B
2557 OFBA 58
                                                                    pop
                                                                    pop
                                             jmp
nop
DATA_BYTE_NOT_READY
2558 OFBB EB 13
                                                                               short WAIT_DONE
2559 OFBD 90
2561 OFBE E8 1028 R
                                                                               BDA_TIMER_TO_CX_DX
                                                                    call
2562 OFC1 5B
2563 OFC2 58
                                                                    рор
                                                                    pop
2564 OFC3 3B D9
                                                                    cmp
                                                                               bx. cx
2565 OFC5 77 E1
2566 OFC7 72 2C
2567 OFC9 3B C2
                                                                               short CHECK_BYTE_READY_FOR_READ
short TIMED_OUT
                                                                    ja
jb
                                                                    cmp
                                                                               short CHECK_BYTE_READY_FOR_READ
short TIMED_OUT
                                                                    ja
jmp
2568 OFCB 77 DB
2569 OFCD EB 26
2570 OFCF 90
                                                                    пор
2571 OFD0
                                             WAIT_DONE:
                                                                               al, 111b
0Ah, al
2572 OFDO BO 07
                                                                    mov
out
                                                                                                                            ; Mask DRQ3
2573 OFD2 E6 OA
2574
2575 OFD4 32 CO
2576 OFD6 BA 0323
2577 OFD9 E8 103A R
                                                                               al, al
dx, IO_PORT_323_DMA_IRQ
                                                                                                                            ; Disable DMA and IRQ
                                                                    call
                                                                               DO NOTHING
2578 OFDC E8 1137 R
                                                                                OUTB_DX_AL
2580 OFDF BA 0320
2581 OFE2 E8 103A R
2582 OFE5 E8 1135 R
                                                                                dx, IO_PORT_320_DATA
                                                                    mov
                                                                    call
                                                                               DO NOTHING
                                                                                INB_AL_DX
                                                                                                                            ; Read the status byte
2583 OFE8 24 02
                                                                    and
                                                                               al, 2
2584 OFEA A2 0474 R
                                                                               BDA_LAST_OP_STATUS, al
2585 OFED 3C 00
2586 OFEF 74 03
                                                                    cmp
                                                                                short DONE_HANDLING_RESPONSE
                                                                    'nе
                                                                                                                            : No error
2587 OFF1 E8 103B R
                                                                    call
                                                                               READ_ERROR_SENSE
                                                                                                                             ; Error encountered, read sense
2588 OFF4
2589 OFF4 C3
                                              DONE_HANDLING_F
                                                                    ret
2590
2591 OFF5
2592 OFF5 BO 07
2593 OFF7 E6 OA
                                             TIMED_OUT:
                                                                                al, 111b
                                                                                                                            ; Mask DRQ3
                                                                    out
                                                                               OAh, al
2594
2595 OFF9 32 CO
                                                                                                                            ; Disable DMA and IRQ
                                                                    xor
                                                                                al, al
dx, IO_PORT_323_DMA_IRQ
2596 OFFB BA 0323
                                                                    mov
2597 OFFE E8 103A R
                                                                    call
2598 1001 E8 1137 R
                                                                               OUTB_DX_AL
                                                                    call
2599
2600 1004 80 3E 0442 R 0C
                                                                    cmp
je
call
                                                                               BDA_CONTROLLER_DATA_BUFFER_00, OP_OC_INIT_DRV_PARM short RETURN_ERROR_BAD_DISK_TABLE
2601 1009 74 0A
2602 100B E8 0CE7 R
                                                                               RESET_CONTROLLER
                                                                               BDA_LAST_OP_STATUS, INT13H_STATUS_80_TIMEOUT
2603 100E C6 06 0474 R 80
2604 1013 EB 08
                                                                                short FINISHED_HANDLING_RESPONSE
                                                                    jmp
2605
2606 1015
                                             RETURN ERROR BAD DISK TABLE:
2607 1015 E8 0D61 R
2608 1018 C6 06 0474 R 07
                                                                              MASTER_RESET
BDA_LAST_OP_STATUS, INT13H_STATUS_07_BAD_DISK_PARAM_TABLE
                                                                    mov
2609
2610 101D
2611 101D C3
                                             FINISHED_HANDLING_RESPONSE:
                                             HANDLE COMMAND RESPONSE ENDP
2612 101E
2613
2614
```

2615			PAGE	
2616			; Just gets the pointer to INT41H.	
2618			;	_
2619			•	
2620	101E		DRIVE_PARAM_TO_DI PROC NEAR	
2621	101E :	LE	push ds	
2622	101F	E8 1141 R	call ZERO_DS_ALT	
2623			ASSUME ds: ZEROSEG	
2624	1022	C4 3E 0104 R	les di, dword ptr INT41H_OFFSET	
2625	1026	LF	pop ds	
2626	1027	23	ret	
2627			DRIVE_PARAM_TO_DI ENDP	
2628				
2629			;	_
2630			; Read the timer. Used for delays.	
2631				_
2632			•	
2633	1028		BDA_TIMER_TO_CX_DX PROC NEAR	
2634	1028	50	push ax	
2635	1029	LE	push ds	
2636	102A	E8 1139 R	call ZERO DS	
2637			ASSUME ds:ZEROSEG	
2638	102D F	FA	cli	
2639	102E 8	BB 16 046C R	mov dx, BDA_TIMER_COUNTER_LO	
2640	1032 8	3B OE 046E R	mov cx, BDA_TIMER_COUNTER_HI	
2641	1036 E	FB.	sti	
2642	1037	LF	pop ds	
2643	1038	58	pop ax	
2644	1039	23	ret	
2645	103A		BDA_TIMER_TO_CX_DX ENDP	
2646				
2647			•	
2648			God knows why does this exist. Perhaps it	
2649			; is makes it easier to hook a tracing routine.	
2650				_
2651			*	
	103A		DO_NOTHING PROC NEAR	
	103A	73	ret	
2654			DO_NOTHING ENDP	
2655	2030		Do-wormen Bude	
-033				

```
2657
                                           PAGE
2658
2659
                                                      This routine is called when a command
2660
                                                      returns an error status.
2661
                                                      It reads in the error details from controler
2662
2663
                                                      and tries to figure out the proper INT13H
                                                      error code.
2664
2665
2666 103B
                                           READ_ERROR_SENSE
                                                                            PROC NEAR
2667 103B E8 1139 R
                                                                call
                                                                           ZERO_DS
2668
                                                                ASSUME
                                                                            ds: ZEROSEG
2669 103E C6 06 0442 R 03
                                                                            BDA_CONTROLLER_DATA_BUFFER_00, OP_03_READ_SENSE
                                                                 call
2670 1043 E8 0F39 R
                                                                            SELECT_BUSY_FOR_COMMAND
2671 1046 E8 0EFC R
2672 1049 72 36
                                                                call
jc
                                                                            WRITE COMMAND BUFFER
2673 104B E8 0F45 R
                                                                 call
                                                                            WAIT FOR BYTE READY
2674 104E 72 31
                                                                            short SENSE_FAILED
                                                                j¢
2675
2676 1050 33 C9
                                                                           cx, cx
si, 0
dx, IO_PORT_320_DATA
DO_NOTHING
                                                                xor
2677 1052 BE 0000
2678 1055 BA 0320
2679 1058 E8 103A R
                                                                mov
                                                                 call
2680 105B
                                           READ_SENSE_BYTE:
2681 105B E8 0F45 R
2682 105E 72 21
2683 1060 E8 1135 R
                                                                            WAIT_FOR_BYTE_READY short SENSE_FAILED
                                                                 call
                                                                 jс
                                                                 call
                                                                            INB AL DX
                                                                            BDA_CONTROLLER_DATA_BUFFER_00[si], al
2684 1063 88 84 0442 R
2685 1067 46
                                                                 inc
                                                                            si
2686 1068 FE C1
                                                                 inc
                                                                            cl
2687 106A 80 F9 04
2688 106D 75 EC
                                                                            short READ_SENSE_BYTE
                                                                jne
2689
2690 106F E8 0F45 R
                                                                call
                                                                           WAIT FOR BYTE READY
2691 1072 72 0D
2692 1074 BA 0320
                                                                            short SENSE_FAILED
dx, IO_PORT_320_DATA
                                                                 j¢
                                                                 mov
2693 1077 E8 103A R
2694 107A E8 1135 R
2695 107D 24 02
                                                                 call
                                                                            DO NOTHING
                                                                            INB_AL_DX
                                                                 and
2696 107F 74 09
                                                                            short XLATE_SENSE_TO_INT13H_STATUS
                                                                 jе
2697 1081
2698 1081 E8 OCE7 R
                                           SENSE_FAILED:
                                                                           RESET_CONTROLLER
                                                                call
2699 1084 C6 06 0474 R FF
                                                                            BDA_LAST_OP_STATUS, INT13H_STATUS_FF_SENSE_OP_FAILED
2700 1089 C3
2701
2702 108A
                                           XLATE SENSE TO INT13H STATUS:
                                                                            b1, BDA_CONTROLLER_DATA_BUFFER_00
2703 108A 8A 1E 0442 R
2704 108E 8A FB
                                                                            bh, bl
2705 1090 B1 04
                                                                mov
                                                                            cl, 4
2706 1092 D2 EB
2707 1094 80 E3 03
                                                                           bl, cl
bl, 3
                                                                 shr
                                                                 and
                                                                            si, offset SENSE ERR Ox
2708 1097 BE 1111 R
                                                                mov
2709 109A 80 FB 00
2710 109D 74 16
2711 109F 81 C6 000A
                                                                 cmp
                                                                            ы, о
                                                                           short DO_XLATE_SENSE
si, SENSE_ERR_1x - SENSE_ERR_0x
                                                                 je
add
2712 10A3 80 FB 01
2713 10A6 74 0D
2714 10A8 81 C6 0010
                                                                 cmp
je
add
                                                                            bl, 1
                                                                           short DO_XLATE_SENSE
si, SENSE_ERR_2x - SENSE_ERR_1x
2715 10AC 80 FB 02
2716 10AF 74 04
2717 10B1 81 C6 0004
                                                                cmp
je
add
                                                                            bl. 2
                                                                           short DO_XLATE_SENSE
si, SENSE_ERR_3x - SENSE_ERR_2x
2718 10B5
                                           DO_XLATE_SENSE:
2719 10B5 80 E7 0F
2720 10B8 53
                                                                 and
                                                                            bh, OFh
                                                                push
                                                                           bx
2721 10B9 8A C7
                                                                 mov
                                                                            al, bh
2722 10BB 32 FF
2723 10BD 2E: 3A 87 1131 R
                                                                 xor
                                                                            al, cs:SENSE_ERR_HI[bx]
                                                                 cmp
                                                                pop
jb
mov
2724 10C2 5B
2725 10C3 72 06
2726 10C5 C6 06 0474 R BB
                                                                           short XLATE_SUCCESSFUL
BDA_LAST_OP_STATUS, INT13H_STATUS_BB_UNDEFINED_ERROR
2727 10CA C3
2728 10CB
                                                                ret
                                           XLATE_SUCCESSFUL:
2728 10CB
2729 10CB 86 FB
2730 10CD 32 FF
2731 10CF 2E: 8A 00
                                                                 xor
                                                                           bh, bh
                                                                            al, cs:[bx+si]
2732 10D2 A2 0474 R
2733 10D5 8A 0E 0442 R
                                                                           BDA_LAST_OP_STATUS, al
cl, BDA_CONTROLLER_DATA_BUFFER_00
                                                                mov
                                                                mov
                                                                            cl, 1Fh
cl, 18h
2734 10D9 80 E1 1F
                                                                 and
2735 10DC 80 F9 18
2736 10DF 75 2F
                                                                 cmp
                                                                                                                     ; 018H = Correctable ECC error
                                                                            short DONE_READING_SENSE
                                                                 jne
                                                                           BDA_CONTROLLER_DATA_BUFFER_00, OP_0D_READ_ECC_BURST_ERROR_LEN SELECT_BUSY_FOR_COMMAND WRITE_COMMAND_BUFFER
2737 10E1 C6 06 0442 R 0D
                                                                 mov
2738 10E1 C6 06 0442
2738 10E6 E8 0F39 R
2739 10E9 E8 0EFC R
2740 10EC 73 07
                                                                 call
                                                                 call
                                          jnb
ERROR_READING_SENSE:
                                                                            short READ ECC COMMAND OKAY
2741 10EE
2742 10EE E8 0CE7 R
                                                                           RESET_CONTROLLER
                                                                call
2743 10F1 F9
                                                                stc
2744 10F2 EB 1C
2745 10F4 90
                                                                            short DONE_READING_SENSE
                                                                пор
2746
2747 10F5
2748 10F5 E8 0F45 R
2749 10F8 72 F4
                                           READ_ECC_COMMAND_OKAY:
                                                                            WAIT_FOR_BYTE_READY
short ERROR_READING_SENSE
                                                                 call
                                                                jç
2750 10FA BA 0320
                                                                            dx, IO_PORT_320_DATA
2751 10FD E8 103A R
2752 1100 E8 1135 R
                                                                           DO_NOTHING
INB_AL_DX
                                                                 call
                                                                call
                                                                push
call
2753 1103 50
2754 1104 E8 0F45 R
2755 1107 73 03
                                                                            WAIT_FOR_BYTE_READY
short READ_ECC_READY
                                                                 jnb
2756 1109 58
2757 110A EB E2
2758
                                                                            short ERROR_READING_SENSE
2759 110C
                                           READ_ECC_READY:
2760 110C E8 1135 R
2761 110F 58
2762 1110
                                                                c=11
                                                                            INB_AL_DX
                                                                pop
                                                                            ax
                                          DONE READING SENSE:
2763 1110 C3
                                          READ_ERROR_SENSE
2764 1111
2765
2766
```

	PAGE		
8 9	Sparse	table fo	or mapping senser error byte :
0			oller to INT13H error code :
1			
2	•		
3 1111	SENSE_ERR_0x:		
4 1111 00		DB	INT13H_STATUS_00_NO_ERROR
5 1112 20		DB	INT13H_STATUS_20_CTRLR_ERROR
6 1113 40		DB	INT13H_STATUS_40_SEEK_FAILURE
7 1114 20		DB	INT13H_STATUS_20_CTRLR_ERROR
8 1115 80		DB	INT13H_STATUS_80_TIMEOUT
9 1116 BB		DB	INT13H_STATUS_BB_UNDEFINED_ERROR
0 1117 20		DB	INT13H_STATUS_20_CTRLR_ERROR
1 1118 BB		DB	INT13H_STATUS_BB_UNDEFINED_ERROR
2 1119 40		DB	INT13H_STATUS_40_SEEK_FAILURE
3 111A 27		DB	INT13H_STATUS_27_NEED_RECALIBRATE
4 111B	SENSE_ERR_1x:		
5 111B 10		DB	INT13H_STATUS_10_ECC_ERROR
6 111C 10 7 111D 02		DB	INTI3H_STATUS_10_ECC_ERROR
		DB	INTI3H_STATUS_02_ADDR_MARK_NOT_FOUND
8 111E 02		DB	INTI3H_STATUS_02_ADDR_MARK_NOT_FOUND
9 111F 04 0 1120 40		DB DB	INT13H_STATUS_04_SECTOR_NOT_FOUND
		DB DB	INT13H_STATUS_40_SEEK_FAILURE
1 1121 20 2 1122 29		DB	INT13H_STATUS_20_CTRLR_ERROR 29h
3 1123 11		DB	INT13H_STATUS_11_ECC_FIXED
4 1124 OB		DB	INTI3H_STATUS_II_ECC_FIXED INTI3H_STATUS_OB_BAD_CYLINDER
5 1125 21		DB	21h
6 1126 BB		DB	INT13H_STATUS_BB_UNDEFINED_ERROR
7 1127 22		DB	22h
8 1128 23		DB	23h
9 1129 24		DB	24h
0 112A 25		DB	25h
1 112B	SENSE_ERR_2x:		
2 112B 01		DB	INT13H_STATUS_01_BAD_COMMAND
3 112C 02		DB	INT13H_STATUS_02_ADDR_MARK_NOT_FOUND
4 112D 01		DB	INT13H_STATUS_01_BAD_COMMAND
5 112E 04		DB	INT13H_STATUS_04_SECTOR_NOT_FOUND
6 112F	SENSE_ERR_3x:		
7 112F 20		DB	INT13H_STATUS_20_CTRLR_ERROR
8 1130 20		DB	INT13H_STATUS_20_CTRLR_ERROR
9 1131	SENSE_ERR_END:		_
0			
1 1131 OA	SENSE_ERR_HI	DB	SENSE_ERR_1x - SENSE_ERR_0x
2 1132 10		DB	SENSE_ERR_2x - SENSE_ERR_1x
3 1133 04		DB	SENSE_ERR_3x - SENSE_ERR_2x
4 1134 02		DB	SENSE_ERR_END - SENSE_ERR_3x
.5			
.6	,		
.7	; I/O rou	tines.	:
.8	;		:
.9			tines seem to make little :
0			call insn is larger than :
1			Perhaps this is useful for :
2	; hooking	up trac	ce routines or something :
3	;		
4	TAID		PROG MEND
5 1135 6 1135 FC	INB_AL_DX	4-	PROC NEAR
6 1135 EC		in	al, dx
7 1136 C3 8 1137	TAID AT DU	ret	ENDP
	INB_AL_DX		ENDE
9 1127	OTHER BY ST		PROC NEAR
0 1137 1 1137 EE	OUTB_DX_AL	out	
		out	dx, al
		ret	ENDP
2 1138 C3	OTIME BY ST		
2 1138 C3 3 1139	OUTB_DX_AL		
2 1138 C3 3 1139 4	OUTB_DX_AL		
2 1138 C3 3 1139 4 5			
2 1138 C3 3 1139 4 5	;		
2 1138 C3 3 1139 4 5 6 6	;; ; Two dif	fferent v	rays to set DS to zero :
2 1138 C3 3 1139 45 5 6 6 7	;; ; Two dif	ferent we why	rays to set DS to zero :
2 1138 C3 3 1139 4 5 6 6 7 7	; Two dif ; Not sur	ferent we why	rays to set DS to zero :
12 1138 C3 3 1139 4.5 5.6 6.7 8.8 9.9	; Two dif ; Not sur ;	ferent we why	rays to set DS to zero :
2 1138 C3 3 1139 4 5 6 6 7 8 8 9 0 1 1139	; Two dif ; Not sur	fferent were why	rays to set DS to zero : :
12 1138 C3 3 1139 4 5 5 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6	; Two dif ; Not sur ;	fferent were why	PROC NEAR
2 1138 C3 3 1139 4 5 6 6 7 8 9 0 1 1139 2 1139 50 3 113A B8 R	; Two dif ; Not sur ;	push	rays to set DS to zero : : PROC NEAR ax ax, ZEROSEG
2 1138 C3 3 1139 4 5 6 6 7 8 9 9 0 1 1139 2 1139 50 3 113A B8 R 4 113D 50	; Two dif ; Not sur ;	push mov push	PROC NEAR ax ax, ZEROSEG
12 1138 C3 13 14 15 16 17 18 19 0 1 1139 2 1139 50 3 113A B8 R 4 113D 50 5 113E IF	; Two dif ; Not sur ;	push mov push pop	PROC NEAR ax ax, ZEROSEG ax ds
12 1138 C3 3 1139 45 56 67 78 8 99 0 1 1139 2 1139 50 3 113A B8 R 4 113D 50 5 113E IF 6 113F 58	; Two dif ; Not sur ;	push mov push pop	PROC NEAR ax ax, ZEROSEG
2 1138 C3 3 1139 4 5 6 6 7 7 8 9 0 1 1139 2 1139 50 3 113A B8 R 4 113D 50 5 113E 1F 6 113F 58 7 1140 C3	; Two dif; Not sur; ZERO_DS	push mov push pop	PROC NEAR ax ax, ZEROSEG ax ds ax
12 1138 C3 13 1139 4 4 5 6 6 7 8 8 9 0 1 1139 2 1139 50 3 113A B8 R 4 113D 50 5 113E 1F 6 113F 58 7 1140 C3 8 1141	; Two dif ; Not sur ;	push mov push pop	PROC NEAR ax ax, ZEROSEG ax ds
22 1138 C3 33 1139 44 45 66 67 77 88 99 00 1 1139 22 1139 50 33 113A B8 R 4 113D 50 5 113E IF 6 113F 58 7 1140 C3 8	; Two did; ; Not sur; ; Tero_DS	push mov push pop	PROC NEAR ax ax, ZEROSEG ax ds ax
22 1138 C3 3 1139 4 1139 4 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	; Two dif; Not sur; ZERO_DS	push mov push pop pop	PROC NEAR ax ax, ZEROSEG ax ds ax ENDP
12 1138 C3 13 1139 14 15 16 17 18 19 0 1 1139 2 1139 50 3 113A B8 R 4 113D 50 5 113E 1F 6 113F 58 7 1140 C3 8 1141 9 10 1141 11 1141 50	; Two did; ; Not sur; ; Tero_DS	push mov push pop pop ret	PROC NEAR ax ax, ZEROSEG ax ENDP PROC NEAR ax
22 1138 C3 33 1139 44 45 66 67 77 88 99 00 11 1139 22 1139 50 33 113A B8 R 41 113D 50 5113E IF 66 113F 58 7 1140 C3 88 1141 9 10 1141 1141 50 22 1142 33 C0	; Two did; ; Not sur; ; Tero_DS	push mov push pop pop ret	PROC NEAR ax ax, ZEROSEG ax ds ax ENDP PROC NEAR ax ax, ax
2 1138 C3 3 1139 4 5 6 6 7 8 9 0 1 1139 2 1139 50 3 113A B8 R 4 113D 50 5 113E IF 6 113F 58 7 1140 C3 8 1141 9 0 1141 1 1141 50 2 1142 33 C0 3 1144 8E D8	; Two did; ; Not sur; ; Sero_DS	push mov push pop pop ret push mov	PROC NEAR ax ax, ZEROSEG ax ds ax ENDP PROC NEAR ax ax, ax ds, ax
2 1138 C3 3 1139 4 5 6 6 7 8 9 0 1 1139 2 1139 50 3 113A B8 R 4 113D 50 5 113E 1F 6 113F 58 7 1140 C3 8 1141 9 0 1141 1 1141 50 2 1142 33 C0 3 1144 8E D8 4 1146 58	; Two did; ; Not sur; ; Sero_DS	push mov push pop pop ret	PROC NEAR ax ax, ZEROSEG ax ds ax ENDP PROC NEAR ax ax, ax
2 1138 C3 3 1139 4 4 5 6 6 7 7 8 8 9 9 0 1 1139 50 3 113A B8 R 4 113D 50 5 113E 1F 6 113F 58 7 1140 C3 8 1141 9 9 1141 1141 50 2 1142 33 C0 3 1144 8E D8 4 1146 58 5 1147 C3	; Two dif; Not sur; ZERO_DS	push mov push pop pop ret push mov	PROC NEAR ax ax, ZEROSEG ax ds ax ENDP PROC NEAR ax ax ax ENDP
2 1138 C3 3 1139 4 5 6 6 7 8 8 9 0 1 1139 2 1139 50 3 113A B8 R 4 113D 50 5 113E 1F 6 113F 58 7 1140 C3 8 1141 9 0 1141 1 1141 50 2 1142 33 C0 3 1144 8E D8 4 1146 58	; Two did; ; Not sur; ; Sero_DS	push mov push pop pop ret	PROC NEAR ax ax, ZEROSEG ax ds ax ENDP PROC NEAR ax ax, ax ds, ax

```
2859
                                          PAGE
2860
2861
2862
                                                    OS Boot Sector load service (INT 19H)
2863
                                                                         PROC NEAR
dx, 0
READ_BOOT_SECTOR
2864 1148
2865 1148 BA 0000
                                          INT19H_HANDLER
                                                                                                        ; Try the floppy
2866 114B E8 1175 R
                                                               call.
2867 114E 73 20
2868 1150 BA 0080
2869 1153 E8 1175 R
                                                                         short DO_THE_BOOT
dx, 80h
                                                               jnb
                                                                                                        ; Try the hard drive
                                                                         READ_BOOT_SECTOR
                                                               call
2870 1156 73 OF
2871 1158
                                                               jnb
                                                                          short CHECK_BOOT_SIGNATURE
                                          BOOT_FAILED:
2872 1158 BE 11AD R
                                                                         si, offset STRING_HARD_DISK_IS_OFF
                                                               mov
                                                                                                         ; strlen boot failed
2873 115B B9 0061
2874 115E 90
                                                               mov
                                                                          cx, 61h
2875 115F OE
                                                               push
                                                                         CS
2876 1160 1F
2878 1161 E8 0B3C R
                                                                         PRINT_CX_CHARS_FROM_DS_SI
                                                               call
2879 1164
2880 1164 90
2881 1165 EB FD
                                          INFINITE_LOOP:
                                                               nop
                                                                         short INFINITE_LOOP
                                                               jmp
2882 1167
                                          CHECK BOOT SIGNATURE:
2883 1167 26: 81 3E 7DFE AA55
2884 116E 75 E8
                                                                         word ptr es:7DFEh, 0AA55h short BOOT_FAILED
                                                               jne
2885
2886 1170
                                          DO THE BOOT:
2887 1170 EA 7C00 ---- R
                                                                         BOOT_SEC
                                                              jmp
2888 1175
                                          INT19H_HANDLER
                                                                         ENDP
2889
2890
2891
                                                    Expects DX=disk number
2892
2893
                                                    Sets carry on error
2894
2895 1175
2896 1175 33 CO
2897 1177 8E CO
                                          READ_BOOT_SECTOR
                                                                         PROC NEAR
                                                                         ax, ax
es, ax
                                                               mov
2898 1179 BB 7C00 R
                                                               mov
                                                                         bx, offset BOOT_SEC
                                                                                                         : Target address
2899 1179 BB 7C00
2899 117C B9 0002
2900 117F
2901 117F 51
                                                                         cx, 2
                                                               mov
                                                                                                         ; Number of attempts
                                          TRY_RESET:
                                                               push
                                                                         CX
2902 1180 B8 0000
2903 1183 CD 13
                                                                         ax, 0
13h
                                                               mov
                                                                                                         ; Reset disk system
2904 1185 59
                                                               pop
jnb
2905 1186 73 OF
2906 1188 80 FC 80
2907 118B 75 04
                                                                         short RESET_SUCCESSFUL ah, 80h
                                                               cmp
                                                               ine
                                                                         short RESET ANOTHER
                                                               stc
2908 118D F9
                                                                         short DONE_READING_BOOT_SECTOR
2910 1190 90
                                                               nop
2911 1191
2912 1191 E2 EC
2913 1193 F9
                                          RESET_ANOTHER:
                                                               loop
                                                                         TRY_RESET
                                                               stc
2914 1194 EB 16
2915 1196 90
2916 1197
                                                                          short DONE_READING_BOOT_SECTOR
                                                               пор
                                          RESET_SUCCESSFUL:
2917 1197 B9 0002
2918 119A
2919 119A 51
                                                               mov
                                                                         сж, 2
                                                                                                         ; Two attempts
                                                               push
2920 119B B9 0001
2921 119E B8 0201
2922 11A1 CD 13
                                                                                                         ; Track 0, sector 1
                                                               mov
                                                                         cx. 0001h
                                                               mov
                                                                         ax,
13h
                                                                              0201h
                                                                                                         ; 02h = read, 01h sectors
                                                              pop
cmp
je
loop
2923 11A3 59
                                                                         CX
2924 11A4 80 FC 00
2925 11A7 74 03
                                                                          ah, 0
                                                                                                         ; Check status
                                                                          short DONE_READING_BOOT_SECTOR
2926 11A9 E2 EF
                                                                         TRY READ SECTOR
2927 11AB F9
2928 11AC
                                          stc
DONE_READING_BOOT_SECTOR:
2929 11AC C3
                                                               ret
2930 11AD
                                          READ_BOOT_SECTOR
                                                                         ENDE
                                          STRING_HARD_DISK_IS_OFF:

DB " Hard disk is off.", ODh, OAh
2932 11AD
2933 11AD 20 20 48 61 72 64
             20 64 69 73 6B 20
69 73 20 6F 66 66
2935
2936
             2E OD OA
" Set HDD switch to on or", 0Dh, 0Ah
2941 72 0D 0A
2942 11DD 20 20 49 6E 73 65
2943 72 74 20 44 4F 53
2944 20 64 69 73 6B 20
2945 69 6E 20 64 72 69
                                                               DB
                                                                          " Insert DOS disk in drive and RESET the System. "
             76 65 20 61 6E 64
20 52 45 53 45 54
20 74 68 65 20 53
2946
2947
2948
2949
2950
             79 73 74 65 6D 2E
2951 120E 0D 0A
                                          STRING_CRLF
                                                              DB
                                                                         ODh, OAh
2952
```

```
2954
2955
                                                                                          "1701"
"-A"
"-B"
"-C"
"-D"
"-E"
2956 1210 31 37 30 31
2957 1214 2D 41
                                                   STRING_1701
                                                   STRING_A
STRING_B
                                                                                                                   ; Controller SRAM error
                                                                             DB
2958 1216 2D 42
2959 1218 2D 43
2960 121A 2D 44
                                                                            DB
DB
                                                                                                                   ; Controller diags error
; Test ready timed out
; Recalibrate failed
                                                   STRING_D
2961 121C 2D 45
                                                  STRING E
                                                                             DB
                                                                                                                   : Reset failed
2962
2963 121E
                                                   INT13H_TRIVIAL_OP_VECTOR:
2964 121E 01
2965 121F 0D70 R
2966 1221 08
2967 1222 0D0C R
                                                                                          INTI3H_OP_01_GET_DISK_STATUS ; INTI3H services that offset INTI3H_01_GET_DISK_STATUS ; do not require INTI3H_OP_08_GET_DRIVE_PARAMS ; talking to the disk offset INTI3H_08_GET_DRIVE_PARAMS ; controller
                                                                             DB
                                                                             DW
2968
2969 1224
                                                   DMA_COMMAND_TABLE:
2970 1224 08
2971 1225 0A
2972 1226 E5
2973 1227 E6
                                                                             DB
                                                                                          OP 08 READ SECTORS
                                                                                                                                         ; These are commands that
; transmit data using DMA
                                                                             DB
DB
                                                                                          OP_0A_WRITE_SECTORS
OP_E5_READ_LONG
                                                                             DB
                                                                                          OP E6 WRITE LONG
2974 1228 OE
2975 1229 OF
2976
                                                                             DB
                                                                                          OP OE READ SECTOR BUFFER
                                                                                          OP_OF_WRITE_SECTOR_BUFFER
2977 122A
                                                   INT13H_OP_TO_COMMAND_BYTE:
2978 122A 00
2979 122B 00
                                                                                          OP_00_TEST_DRIVE_READY
OP_00_TEST_DRIVE_READY
                                                                                                                                                00h
01h
                                                                             DB
DB
2980 122C 08
                                                                             DB
                                                                                          OP 08 READ SECTORS
                                                                                                                                                 02h
                                                                                          OP_08_READ_SECTORS
OP_08_WRITE_SECTORS
OP_05_VERIFY_SECTORS
2981 122D 0A
2982 122E 05
                                                                             DB
DB
                                                                                                                                                 03h
                                                                                                                                                 04h
2983 122F 06
                                                                             DB
                                                                                          OP 06 FORMAT TRACK
                                                                                                                                                 05h
                                                                                          OP_06_FORMAT_TRACK
OP_07_FORMAT_BAD_TRACK
OP_04_FORMAT_DRIVE
                                                                             DB
DB
DB
2984 1230 07
2985 1231 04
                                                                                                                                                06h
07h
                                                                                          OP_00_TEST_DRIVE_READY
2986 1232 00
                                                                                                                                                 08h
2987 1233 OC
2988 1234 E5
2989 1235 E6
                                                                            DB
DB
DB
                                                                                          OP_OC_INIT_DRV_PARM
OP_E5_READ_LONG
                                                                                                                                                09h
0Ah
                                                                                          OP_E6_WRITE_LONG
                                                                                                                                                 0Bh
                                                                             DB
DB
DB
2990 1236 OB
                                                                                          OP OB SEEK
                                                                                                                                                 0Ch
2991 1237 00
2992 1238 0E
                                                                                          OP_00_TEST_DRIVE_READY
                                                                                                                                                0Dh
0Eh
                                                                                          OP_OE_READ_SECTOR_BUFFER
2993 1239 OF
                                                                             DB
                                                                                          OP OF WRITE SECTOR BUFFER
                                                                                                                                                OFh
2994 123A 00
2995 123B 01
                                                                             DB
DB
                                                                                          OP_00_TEST_DRIVE_READY
OP_01_RECALIBRATE
                                                                                                                                                 10h
                                                                                                                                                 11h
                                                                                          OP_EO_SECTOR_BUFFER_DIAG
2996 123C E0
                                                                             DB
                                                                                                                                                12h
2997 123D E3
2998 123E E4
                                                                             DB
DB
                                                                                         OP_E3_DRIVE_DIAG
OP_E4_CTRL_DIAG
                                                                                                                                                13h
14h
2999
3000 123F 0DB1 [
3001
                                                   PADDING
                                                                             DB
                                                                                         0DB1h dup(0FFh)
3002
                                   ]
3003
3004 1FF0 32 37 39 33 56 47
3005 2E 31 30 30 31 30
3006 36 38 38
3007 1FFF 94
                                                   VERSION_TAG
                                                                                          "2793VG.10010688"
3008
3009 2000
                                                   ROM
                                                                             ENDS
                                                                             END
3011
```

```
; This has been disassembled and commented manually on :
; Sat Sep 11th 2021 in hope it will be useful to :
; somebody. When assembled, it matches the :
Victor V86P Hard Drive Controller ROM versioned :
; "2793VG.10010688" byte-for-byte. :
; Please note that it is not free software and is ; subject to copyright. :
; SMOKE WEED
```